

# ***CLUSTERING TITIK PANAS BUMI MENGGUNAKAN ALGORITME AFFINITY PROPAGATION***

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Barik Kresna Amijaya  
NIM: 125150207111055



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

# PENGESAHAN

## CLUSTERING TITIK PANAS BUMI MENGGUNAKAN ALGORITME AFFINITY PROPAGATION

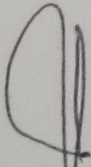
### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Barik Kresna Amijaya  
NIM: 125150207111055

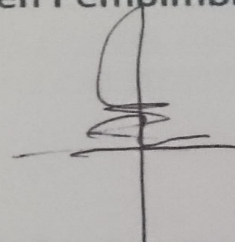
Skripsi ini telah diuji dan dinyatakan lulus pada  
16 Januari 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



M. Tanzil Furqon, S.kom, M.CompSc  
NIP: 19820930 200801 1 004

Dosen Pembimbing II



Candra Dewi, S.Kom, M.Sc  
NIP: 19771114 200312 2 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 16 Januari 2018



Barik Kresna Amijaya

NIM:125150207111055

## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah memberikan kelancaran pada penulisan dalam menyelesaikan skripsi ini sebagai salah satu syarat untuk menyelesaikan studi di Program Studi Teknik Informatika Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.

Dalam penulisan skripsi ini, tertera banyak pihak yang telah memberikan bantuan baik moril maupun petunjuk, dan bimbingan. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. M. Tanzil Furqon, S.Kom, M.CompSc selaku Dosen Pembimbing I yang telah memberikan bimbingan, arahan, dan nasehat untuk kesempurnaan tugas akhir ini.
2. Candra Dewi, S.Kom, M.Sc selaku Dosen Pembimbing II yang telah memberikan bimbingan, arahan, dan nasehat untuk kesempurnaan tugas akhir ini.
3. Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer.
4. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
5. Kedua orang tua penulis yang tidak pernah lelah untuk mendidik, mengajarkan, membantu, menasehati, mendorong penulis untuk bekerja keras dan selalu mendoakan agar menjadi orang baik dan sukses.
6. Penulis juga mengucapkan kepada teman-teman yang telah meluangkan waktu untuk memberi bantuan menyelesaikan tugas akhir ini.
7. Teman – teman seperjuangan yang mengambil data sama dengan dosen pembimbing sama yang selalu menghibur selama pengerjakan tugas akhir dan memberikan ide-ide yang luar biasa.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, maka saran dan kritik yang konstruktif dari semua pihak sangat diharapkan demi menyempurnakan skripsi ini dan semoga skripsi ini bisa memberikan manfaat untuk penelitian selanjutnya.

Akhirnya hanya kepada Tuhan Yang Maha Esa kita kembalikan semua urusan dan semoga skripsi ini dapat bermanfaat bagi semua pihak, khususnya bagi penulis dan para pembaca pada umumnya.

Malang, 16 Januari 2018

Penulis  
barik.kresna24@gmail.com

## ABSTRAK

Kebakaran hutan dan lahan merupakan bencana yang mengakibatkan kerugian yang sangat besar serta dapat mengganggu aktivitas makhluk hidup disekitar lokasi kebakaran. Kebakaran hutan dan lahan seharusnya dapat dicegah dengan cara mengetahui penyebab terjadinya kebakaran. Salah satu cara pencegahan kebakaran adalah melakukan pemantauan titik panas bumi. Titik panas bumi adalah suatu area yang relatif suhunya lebih tinggi jika dibandingkan dengan area sekitarnya yang dideteksi satelit. Area tersebut diwakilkan dalam suatu titik yang memiliki koordinat tertentu. Titik panas bumi perlu dilakukan pengelompokan atau *clustering* agar diketahui kemiripan tiap titik dan mudah untuk dilakukan pemantauan. *Clustering* adalah proses mengelompokkan data ke dalam *cluster*, sehingga objek yang ada didalam suatu *cluster* memiliki kesamaan yang tinggi satu sama lain dan sangat berbeda dengan objek yang ada pada *cluster* lain. Metode *Affinity Propagation* merupakan metode yang digunakan untuk melakukan pengelompokkan data dengan menentukan eksemplar sebagai pusat data. *Affinity Propagation* melakukan *clustering* dengan mencari nilai *responsibility* dan *availability* tiap data sehingga menemukan eksemplar yang tepat. Dalam penelitian ini telah dilakukan *clustering* menggunakan *Affinity Propagation* dengan nilai *silhouette coefficient* terbaik yaitu 0.317818 dengan 125 data dan terbentuk 44 *cluster*.

Kata kunci: Titik panas(*hotspot*), *data mining*, *clustering*, *Affinity Propagation*, *Silhouette Coefficient*

## ABSTRACT

*Forest and land fires are catastrophic and can disrupt the activity of living things around the fire location. Forest and land fires should be prevented by knowing the cause of the fire. One of the ways of fire prevention is to monitor the hotspot. The hotspot is an area where the temperature is relatively higher compared to the area around which the satellite is detected. The area is represented in a point that has certain coordinates. hotspot needs to be grouped or clustered to know the similarity of each point and easy to do monitoring. Clustering is the process of grouping data into clusters, so that objects that exist within a cluster have a high similarity with each other and very different from the objects that exist in other clusters. Affinity Propagation method is a method used to perform data grouping by specifying the exemplar as data centers. Affinity Propagation performs clustering by searching for responsibility value and availability of each data to find the right exemplar. In this research has done clustering using Affinity Propagation with the best silhouette coefficient value that is 0.317818 with 125 data and formed 44 cluster.*

*keywords: hotspot, data mining, clustering, Affinity Propagation, Silhouette Coefficient*

## DAFTAR ISI

PENGESAHAN.....	I
PERNYATAAN ORISINALITAS .....	II
KATA PENGANTAR.....	III
ABSTRAK.....	IV
ABSTRACT .....	V
DAFTAR ISI .....	VI
DAFTAR TABEL.....	VIII
DAFTAR GAMBAR .....	IX
DAFTAR KODE PROGRAM .....	X
DAFTAR PERSAMAAN .....	XI
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Manfaat .....	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Penulisan .....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	4
2.1 Tinjauan Pustaka.....	4
2.2 Data Mining .....	6
2.3 Clustering.....	7
2.4 Metode Affinity Propagation.....	7
2.5 Titik Panas Bumi (Hotspot) .....	8
2.6 Silhouette Coefficient .....	9
BAB 3 METODOLOGI.....	10
3.1 Metodologi Penelitian .....	10
3.2 Studi Literatur .....	11
3.3 Pengumpulan Data .....	11
3.4 Analisis dan Perancangan .....	11
3.5 Implementasi .....	11
3.6 Pengujian dan Analisis .....	12
3.7 Evaluasi dan Analisis Hasil .....	12
BAB 4 PERANCANGAN.....	13
4.1 Deskripsi Sistem.....	13
4.2 Perancangan Perangkat Lunak .....	13
4.2.1 Affinity Propagation.....	14
4.3 Perhitungan Manual .....	16
4.3.1 Affinity Propagation.....	16
4.3.2 Silhouette Coefficient .....	23
4.4 Desain Antarmuka Aplikasi .....	27
4.4.1 Perancangan Antarmuka Data.....	27

4.4.2 Perancangan Antarmuka Similarity .....	28
4.4.3 Perancangan Antarmuka Responsibility.....	29
4.4.4 Perancangan Antarmuka Availability .....	30
4.4.5 Perancangan Antarmuka Sum Responsibility dan Availability .	31
4.4.6 Perancangan Antarmuka Iterasi .....	32
4.4.7 Perancangan Antarmuka <i>Cluster</i> .....	33
BAB 5 IMPLEMENTASI .....	34
5.1 Implementasi Algoritme .....	34
5.1.1 Proses Algoritme Similarity .....	34
5.1.2 Proses Algoritme Responsibility .....	35
5.1.3 Proses Algoritme Availability .....	36
5.1.4 Proses Algoritme Penjumlahan Responsibility dan Availability	37
5.1.5 Proses Algoritme Pengelompokan <i>Cluster</i> .....	38
5.1.6 Proses Algoritme Silhouette Coefficient .....	40
5.2 Implementasi Antarmuka .....	42
5.2.1 Implementasi Antarmuka Data.....	42
5.2.2 Implementasi Antarmuka Similarity .....	43
5.2.3 Implementasi Antarmuka Responsibility.....	43
5.2.4 Implementasi Antarmuka Availability .....	44
5.2.5 Implementasi Antarmuka Sum Responsibility Availability .....	44
5.2.6 Implementasi Antarmuka Iterasi .....	45
5.2.7 Implementasi Antarmuka <i>Cluster</i> .....	45
5.2.8 Implementasi Antarmuka Silhouette Coefficient .....	46
BAB 6 PENGUJIAN.....	47
6.1 Pengujian Kualitas <i>Cluster</i> Berdasarkan Jumlah Data .....	47
6.1.1 Pengujian <i>Cluster</i> Berdasarkan Jumlah Data.....	47
6.1.2 Analisis Pengujian Kualitas <i>Cluster</i> .....	49
BAB 7 PENUTUP .....	50
7.1 Kesimpulan .....	50
7.2 Saran .....	50
DAFTAR PUSTAKA.....	51
LAMPIRAN 44 <i>CLUSTER</i> .....	53



## DAFTAR TABEL

Tabel 2. 1 Tabel perbandingan pustaka .....	5
Tabel 4. 1 data panas bumi .....	16
Tabel 4. 2 perhitungan <i>Similarity</i> tiap titik data .....	17
Tabel 4. 3 Perhitungan <i>responsibility</i> tiap titik data .....	18
Tabel 4. 4 perhitungan <i>self-responsibility</i> tiap titik data .....	19
Tabel 4. 5 perhitungan <i>availability</i> tiap titik data .....	20
Tabel 4. 6 perhitungan <i>self-availability</i> tiap titik data .....	21
Tabel 4. 7 perhitungan responsibility + availability .....	22
Tabel 4. 8 pembentukan <i>cluster</i> .....	23
Tabel 4. 9 hasil $x(i)$ .....	25
Tabel 4. 10 hasil $y(i)$ .....	25
Tabel 4. 11 hasil $s(i)$ .....	26
Tabel 4. 12 Penjelasan Antarmuka Data .....	27
Tabel 4. 13 Penjelasan Rancang Antarmuka Similarity .....	28
Tabel 4. 14 Penjelasan Rancang Antarmuka Responsibility .....	29
Tabel 4. 15 Penjelasan Antarmuka Availability .....	30
Tabel 4. 16 penjelasan Antarmuka Sum Responsibility dan Availability .....	31
Tabel 4. 17 Penjelasan Antarmuka Iterasi .....	32
Tabel 4. 18 Penjelasan Rancang Antarmuka <i>Cluster</i> .....	33
Tabel 6. 1 pengujian .....	47
Tabel 6. 2 nilai correlation coefficient .....	49

## DAFTAR GAMBAR

Gambar 2. 1 Langkah Data Mining.....	6
Gambar 3. 1 Diagram Alir Metodologi.....	10
Gambar 4. 1 Diagram Alir Perancangan Sistem.....	13
Gambar 4. 2 Diagram Alur Perancangan.....	14
Gambar 4. 3 flowchart affinity propagation .....	15
Gambar 4. 4 Flowchart Silhouette Coefficient.....	24
Gambar 4. 5 Rancang Antarmuka Data.....	27
Gambar 4. 6 Rancang Antarmuka Similarity .....	28
Gambar 4. 7 Rancang Antarmuka Responsibility.....	29
Gambar 4. 8 Rancang Antarmuka Availability .....	30
Gambar 4. 9 Rancang Antarmuka Sum Responsibility dan Availability.....	31
Gambar 4. 10 Rancang Antarmuka Iterasi .....	32
Gambar 4. 11 Rancang Antarmuka <i>Cluster</i> .....	33
Gambar 5. 1 Diagram Implmentasi.....	34
Gambar 5. 2 Tampilan Halaman Data .....	42
Gambar 5. 3 Tampilan Halaman Similarity .....	43
Gambar 5. 4 Tampilan Halaman Responsibility .....	43
Gambar 5. 5 Tampilan Halaman Availability.....	44
Gambar 5. 6 Tampilan Halaman Responsi Availability .....	44
Gambar 5. 7 Tampilan Halaman Iterasi .....	45
Gambar 5. 8 Tampilan Halaman <i>Cluster</i> .....	45
Gambar 5. 9 Tampilan Halaman Silhouette Coefficient .....	46
Gambar 6. 1 Grafik pengaruh perbedaan data latih terhadap <i>cluster</i> yang terbentuk.....	48
Gambar 6. 2 grafik pengaruh perbedaan jumlah data terhadap silhoette coefficient.....	48

## DAFTAR KODE PROGRAM

Kode Program 5. 1 kode program similarity .....	34
Kode Program 5. 2 kode program responsibility .....	36
Kode Program 5. 3 kode program availability.....	37
Kode Program 5. 4 Source Code Sum Responsibility Availability .....	38
Kode Program 5. 5 kode program pembentukan <i>cluster</i> .....	40
Kode Program 5. 6 kode program Silhouette Coefficient.....	41

## DAFTAR PERSAMAAN

Persamaan (2.1).....	8
Persamaan (2.2).....	8
Persamaan (2.3).....	8
Persamaan (2.4).....	9

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Kita sering membaca *running text* suatu berita di televisi seperti “Satelit NOAA telah mendeteksi 178 titik api di Provinsi Riau”. Istilah titik api yang digunakan sebagai deteksi dari Satelit NOAA adalah titik panas atau *hotspot*. *Hotspot* adalah suatu area yang relatif suhunya lebih tinggi jika dibandingkan dengan area sekitarnya yang dideteksi satelit. Area tersebut diwakilkan dalam suatu titik yang memiliki koordinat tertentu (lapan, 2014).

Dari keterangan di atas, keberadaan titik panas mengindikasikan adanya suhu tinggi yang bisa jadi berpotensi kebakaran atau bisa juga titik panas tersebut hanya menunjukkan suhu tinggi tanpa potensi kebakaran. Oleh karena itu dilakukan *clustering* untuk menentukan apakah titik panas di area tertentu berpotensi kebakaran atau tidak dan dapat dilakukan tindakan pencegahan agar tidak menimbulkan kerugian yang lebih besar. *Clustering* merupakan alat *fundamental* untuk analisis data yang bertujuan untuk mengidentifikasi beberapa struktur *inheren* yang ada pada satu set objek. Kita menggunakan algoritme *clustering* untuk secara otomatis membagi satu set objek atau titik data menjadi beberapa kelompok data sehingga objek yang sama berada dalam kelompok yang sama atau mirip satu sama lain, sementara objek dari kelompok yang lain berbeda (Guojun, 2014). Dalam penelitian ini akan menggunakan metode *clustering Affinity Propagation*.

Metode *Affinity Propagation* yang mengambil masukan ukuran kesamaan antara pasangan titik data, pesan bernilai *real* di pertukarkan antara titik data sampai satu set eksemplar berkualitas tinggi dan *cluster* yang sesuai secara bertahap muncul, keunggulannya adalah menemukan *cluster* dengan kesalahan jauh lebih rendah dari metode lain dan dilakukan dengan waktu yang relatif singkat. Penelitian sebelumnya mengenai *Affinity Propagation* sudah pernah dilakukan oleh Brendan J. Frey dan Delbert Dueck dan hasil yang didapat adalah 62 *cluster* terbentuk dari *clustering* 900 wajah dengan rata-rata *squared error* 108 (Brendan and Delbert, 2007).

Penelitian lain dilakukan terkait sebaran titik panas adalah deteksi pencilan titik panas yang dilakukan oleh Baehaki, Deteksi pencilan dilakukan dengan menggunakan metode *clustering* k-means serta digunakan pendekatan pencilan global dan kolektif. menggunakan nilai k sebesar 10 Pada fungsi k-means menghasilkan nilai *error* 18 526.14. Hasil perhitungan menunjukkan bahwa berdasarkan pendekatan kolektif terdeteksi pencilan titik api muncul pada beberapa *cluster* yaitu 5, 7, dan 10. Di samping itu diperoleh sebanyak 30 pencilan titik api yang berdasarkan pendekatan global. Kemunculan pencilan pada data titik api terjadi pada bulan Februari, Maret, Juni, dan Agustus. Frekuensi pencilan titik api tertinggi terjadi pada 21 Juni 2005 adalah sebanyak 1118. Adapun frekuensi kemunculan pencilan titik api terkecil adalah 295 titik api



dan rata-rata frekuensi kemunculan pencilan titik api sebesar 482.22 (Baehaki, 2014).

Dari permasalahan dan penjelasan dari beberapa penelitian mengenai titik panas bumi dan *Affinity Propagation*, maka peneliti akan mencoba mengimplementasikan metode *Affinity Propagation*. Oleh karena itu judul yang diambil untuk skripsi yaitu "*CLUSTERING* TITIK PANAS BUMI MENGGUNAKAN ALGORITME *AFFINITY PROPAGATION*".

## 1.2 Rumusan Masalah

Mengacu pada latar belakang yang telah diutarakan di atas, maka perlu dirumuskan agar masalah pada penelitian ini lebih terarah dan juga sesuai dengan tujuan. Rumusan masalah dalam penelitian ini antara lain:

1. Bagaimana meng*cluster* data titik panas bumi?
2. Bagaimana kualitas dari hasil *cluster* pada data titik panas bumi?

## 1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini antara lain:

1. Menerapkan metode *Affinity Propagation* untuk *clustering* data titik panas bumi.
2. Mengetahui kualitas dari hasil *cluster* pada data titik panas bumi menggunakan *Silhouette Coefficient*.

## 1.4 Manfaat

Hasil penelitian ini yaitu deteksi dini kemunculan titik api dan diharapkan dapat mempermudah pihak terkait dalam pengambilan keputusan dan upaya pencegahan kebakaran hutan.

## 1.5 Batasan Masalah

supaya penelitian ini lebih terfokus dan tidak terlalu luas pada pembahasannya, maka perlu diberikan batasan masalah. Adapun batasan masalah dalam penelitian ini adalah:

1. Data yang digunakan dalam penelitian ini diperoleh dari *National Aeronautics and Space Administration* (NASA) yang berupa data titik panas bumi bagian Asia Tenggara pada jenjang waktu 7 hari yaitu 18 Agustus 2016 sampai dengan 24 Agustus 2016.
2. Fitur yang akan digunakan dalam aplikasi adalah *brightness*, *FRP*, dan *confidence*.

## **1.6 Sistematika Penulisan**

### **BAB I PENDAHULUAN**

pada bab ini diuraikan latar belakang penelitian, tujuan penelitian, rumusan masalah, batasan masalah, dan manfaat penelitian serta sistematika penulisan.

### **BAB II LANDASAN KEPUSTAKAAN**

Dalam bab ini menguraikan tentang dasar teori dan referensi yang mendasari proses perancangan dan implementasi penelitian ini.

### **BAB III METODOLOGI**

Bab ini merupakan inti dari penelitian, berisi tentang perancangan sistem serta batasan dari perancangan program.

### **BAB IV PERANCANGAN**

Di dalam bab ini data atau informasi hasil penelitian diolah, Perhitungan manual dan perancangan antarmuka.

### **BAB V IMPLEMENTASI**

Dalam bab ini menguraikan bagaimana peneliti mengimplementasikan algoritme menggunakan bahasa pemrograman.

### **BAB VI PENGUJIAN**

Bab ini menguraikan bagaimana hasil implementasi akan diuji.

### **BAB VII PENUTUP**

Bab ini merupakan hasil kesimpulan dari semua yang telah dicapai pada masing-masing bab sebelumnya. Tersusun atas Kesimpulan dan Saran.

## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas tentang landasan pustaka yang mencakup kajian pustaka dan dasar teori yang diperlukan untuk penelitian. landasan pustaka membahas penelitian yang telah dilakukan sebelumnya. Dasar teori adalah membahas teori yang diperlukan untuk menyusun penelitian.

### 2.1 Tinjauan Pustaka

Penelitian pertama yang dibahas, yaitu "*AFFINITY PROPAGATION: CLUSTERING DATA BY PASSING MESSAGES*". *Affintiy Propagation* terbukti menghasilkan *squared error* lebih sedikit daripada metode lainnya. penelitian ini menggunakan metode *Affinity Propagation* untuk *clustering* 900 wajah dan dihasilkan 62 *cluster* dengan rata-rata *squared error* 108 (Delbert, 2007).

Penelitian lain yang dilakukan Xiangliang Zhang "*Data Streaming with Affinity Propagation*" Kontribusi utama dari makalah ini adalah untuk memperpanjang algoritme *Affinity Propagation* yang diusulkan oleh Frey dan Dueck. Perpanjangan ini dienkapsulasi dalam algoritme *Streaming Affinity Propagation*, telah divalidasi secara empiris dalam konteks *streaming* data, pada dua kumpulan data berukuran besar termasuk kumpulan data *Intrusion Detection KDD99*, dan dibandingkan dengan algoritme *state-of-the art DenStream*. Keakuratan model data yang dihasilkan oleh *StrAP* ternyata relatif memuaskan, waktu komputasi lebih tinggi daripada *DenStream* (xiangliang, 2008).

Sama halnya dengan paragraph diatas Guojun Gan melakukan penambahan algoritme pada *Affinity Propagation* dengan penelitiannya "*Subspace clustering using Affinity Propagation*". Dengan memasukkan bobot atribut dalam perhitungan jarak, Guojun memperluas algoritme *Affinity Propagation* untuk mengidentifikasi *cluster* yang tertanam di subruang (*subspace*) dalam ruang data. Besar dari bobot atribut yang relatif dapat digunakan untuk mengidentifikasi dimensi *cluster* subspace. Percobaan pada data sintetis telah menunjukkan bahwa algoritme ini efektif dalam menemukan *cluster* yang tertanam di subruang (*subspace*) dan mengidentifikasi atribut penting (guojun, 2014).

Penelitian lain tentang titik panas dilakukan oleh Sitanggang "*Association Rules Mining untuk Data Kebakaran Hutan Menggunakan Algoritme ECLAT dan SPADE*". algoritme *Equivalence Class Transformation* (ECLAT) dan *sequential Pattern Discovery using Equivalence classes* (SPADE) digunakan pada teknik association rule mining yang pada penelitian ini digunakan untuk menentukan pola keterkaitan antara kemunculan titik api dan faktor-faktor pendukung lainnya (Sitanggang, 2013).

Penelitian lain tentang titik panas dilakukan oleh Adzkia "Deteksi Pencilan Kontekstual pada Data Titik Panas di Provinsi Riau menggunakan Algoritme K-Medoids". Penelitian ini bertujuan mendeteksi pencilan kontekstual pada data

titik panas di Provinsi Riau dari tahun 2002 sampai 2014 berdasarkan konteks cuaca yaitu curah hujan. Pencilan kontekstual dideteksi menggunakan hasil *clustering* pada atribut frekuensi titik panas harian dan curah hujan. Metode yang digunakan adalah teknik *clustering* dengan algoritme k-medoids. Hasil menunjukkan bahwa jumlah pencilan kontekstual yang terdeteksi sebanyak 88 objek (Adzkia, 2016).

**Tabel 2. 1 Tabel perbandingan pustaka**

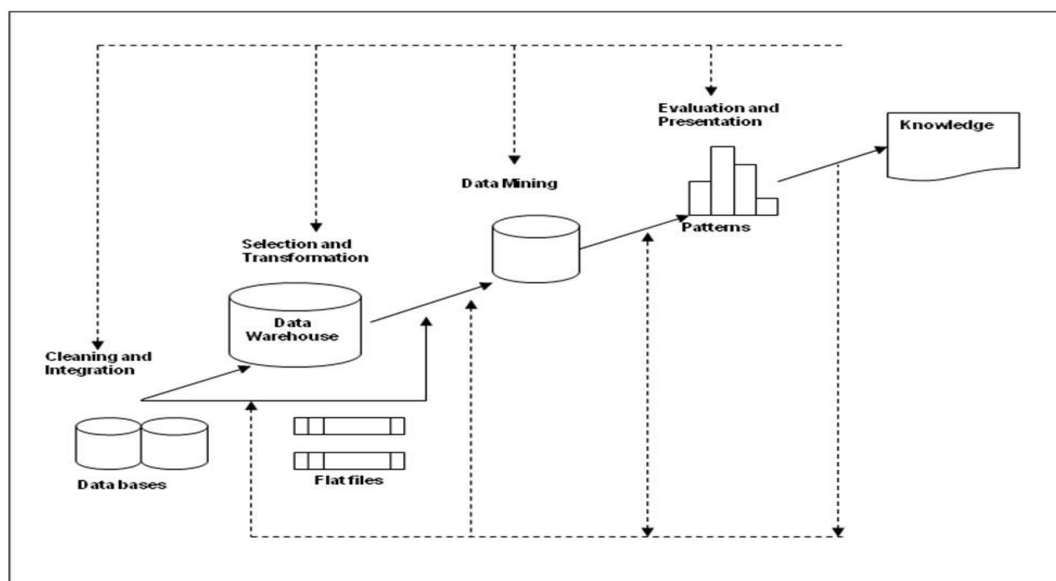
Judul	metode	topik	hasil
<i>Clustering by Passing Messages Between Data Points.</i>	Affinity Propagation.	<i>Clustering</i> 900 wajah.	62 <i>cluster</i> terbentuk dari <i>clustering</i> 900 wajah dengan rata-rata <i>squared error</i> 108.
<i>Data Streaming with Affinity Propagation.</i>	Streaming Affinity Propagation.	<i>Clustering</i> 800 data sintetis dan 1000 <i>intrusion dataset</i> (KDD99).	Mengelompokkan aliran data <i>Intrusion Detection</i> dengan tingkat kesalahan yang sangat rendah kurang dari 1%.
<i>Subspace clustering using Affinity Propagation.</i>	Subspace Affinity Propagation.	<i>Clustering</i> 2000 <i>dimensional dataset points</i> .	Terbentuk 4 <i>cluster</i> dengan beragam <i>subspace dimensions</i> .
<i>Association Rules Mining</i> untuk Data Kebakaran Hutan Menggunakan Algoritme <i>ECLAT</i> dan <i>SPADE</i> .	<i>ECLAT</i> dan <i>SPADE</i> .	Menentukan pola keterkaitan antar wilayah kebakaran di propinsi riau.	pola keterkaitan antar wilayah kebakaran di propinsi riau.
Deteksi Pencilan Kontekstual pada Data Titik Panas di Provinsi Riau menggunakan Algoritme <i>K-Medoids</i> .	<i>K-Medoids</i> .	<i>Clustering</i> data titik panas propinsi riau tahun 2002 sampai 2014.	terdeteksi pencilan kontekstual sebanyak 88 objek.

## 2.2 Data Mining

Pengambilan keputusan dalam ruang lingkup yang lebih kompleks seperti dalam teknik sipil, sangat tergantung terhadap kemampuan kognitif seorang pengambil keputusan itu sendiri, maka secara langsung dapat berdampak terhadap kualitas keputusannya. Saat ini, dengan menggunakan dukungan teknik komputer dan algoritme, model interpretasi dan dukungan pengambilan keputusan sudah berkembang dengan pesat (Cortez, 2010). Seluruh data harus menjadi bahan pertimbangan dalam pengambilan keputusan dengan berbagai metode yang telah ada sekarang secara umum. Data tersebut akan menjadi informasi yang berguna apabila mampu diinterpretasikan secara tepat. Model interpretasi data tersebut selanjutnya disebut *Data Mining* (Sigit, 2015).

Data mining adalah proses dimana metode cerdas diterapkan untuk mendapatkan pola data. Data mining adalah gabungan dari berbagai bidang ilmu, antara lain basis data, *information retrieval*, statistika, *machine learning* dan lain sebagainya. data mining sangat berguna untuk menentukan suatu pola yang didapatkan dari fungsi data mining (Han, J., Kamber, M., dan Pei, J., 2006).

Tujuan dari data mining sangat beragam mulai dari pengklasifikasian, pengelompokan, pencarian, peringkasan dokumen dan lain sebagainya. Dokumen teks merupakan salah satu bentuk dokumen yang sering ditambah atau digunakan. Proses penambangan data sendiri bukan merupakan sebuah proses yang tunggal dan sederhana tetapi merupakan proses yang berkelanjutan dan lebih kompleks, dimulai dari adanya data mentah yang dilakukan proses awal dan diikuti dengan proses penambangan data yang menghasilkan keluaran yang diharapkan (Raharjo, 2014). langkah-langkah untuk melakukan data mining bisa dilihat pada Gambar 2.1.



**Gambar 2. 1 Langkah Data Mining**

Sumber : tinjauan pustaka



## 2.3 Clustering

*Clustering* adalah alat *fundamental* untuk analisis data yang telah luas digunakan dalam aplikasi di banyak bidang teknik dan ilmiah termasuk pengenalan pola, ekstraksi fitur, kuantisasi vektor, segmentasi citra, bioinformatika, dan data mining (Ke-Lin Du, 2013). *Clustering* adalah klasifikasi yang belum ditentukan kelasnya (*cluster*). Dalam proses *clustering*, sampel dibagi berdasarkan kategori yang tiap anggotanya mempunyai kemiripan dan disebut *cluster* (Kruegel, 2005).

*Clustering* adalah proses mengelompokkan data ke dalam *cluster*, sehingga objek yang ada didalam suatu *cluster* memiliki kesamaan yang tinggi satu sama lain dan sangat berbeda dengan objek yang ada pada *cluster* lain. Ketidaksamaan dinilai dari nilai atribut yang menggambarkan suatu objek. Seringkali, ukuran jarak yang digunakan (Han, J., Kamber, M., dan Pei, J., 2006).

## 2.4 Metode Affinity Propagation

Metode ini dimulai dengan mengukur kesamaan antara pasangan titik data dan membuat *passing messages* bernilai *real* antara titik-titik data sampai eksemplas berkualitas tinggi ditemukan dan *cluster* yang sesuai ditemukan. metode *Affinity Propagation* mengansumsikan semua titik data sebagai pusat *cluster* dan tidak tergantung pada kualitas set awal pusat *cluster* (Delbert, 2007). Langkah langkah metode *Affinity Propagation* sebagai berikut

Asumsi  $x_1..x_n$  menjadi satu set titik data, dan  $S$  menjadi fungsi yang mengkuantifikasikan kesamaan antara dua titik, sehingga  $s(X_i, X_j) > s(X_i, X_k)$  jika  $X_i$  lebih mirip dengan  $X_j$  daripada  $X_k$

Algoritme dimulai dengan langkah *message passing*. Untuk *update* dua matriks:

- “*responsibility*” matriks  $R$  memiliki nilai  $r(i,k)$  yang mengukur seberapa cocok  $X_k$  untuk dijadikan eksemplar untuk  $X_i$ .
- “*availability*” matriks  $A$  berisi nilai  $a(i,k)$  menunjukkan seberapa layak untuk  $X_i$  untuk menjadikan  $X_k$  sebagai eksemplarnya.

Kedua matriks diinisialisasi ke nol semua, Algoritme kemudian melakukan pembaruan *iterative* sebagai berikut berikut:

- Pertama, update *responsibility* seperti dibawah ini:  
$$r(i, k) \leftarrow s(i, k) - \max_{k' \neq k} \{a(i, k') + s(i, k')\} \quad (2.1)$$

- Lalu, update *Availability* seperti dibawah ini:  
$$a(i, k) \leftarrow \min \{0, r, (k, k) + \sum_{i' \notin \{i, k\}} \max (0, r(i', k))\}$$
  
untuk  $i \neq k$  (2.2)

Dan,

$$a(k, k) \leftarrow \sum_{i' \neq k} \max (0, r(i', k)) \quad (2.3)$$

Keterangan : *r* = *responsibility*  
*a* = *availability*

## 2.5 Titik Panas Bumi (Hotspot)

Panas bumi dalam penelitian ini adalah titik api yang ada dipermukaan bumi biasanya disebut *burned area*. *Burned area* adalah daerah yang terkena api dalam interval waktu tertentu, deteksi berdasarkan efek api pada vegetasi seperti pembuangan aktivitas fotosintesis, arang di tanah, lahan terbuka (Boschetti, Roy et al, 2008). *Hotspot* dapat diartikan sebagai daerah yang memiliki suhu permukaan relatif lebih tinggi dibandingkan dengan daerah lain di sekitarnya yang berdasarkan ambang batas suhu tertentu yang terpantau satelit penginderaan jarak jauh. Tipologinya adalah titik dan dihitung sebagai satuan jumlah bukan satuan luasan. *Hotspot* adalah hasil deteksi kebakaran hutan/lahan pada ukuran piksel tertentu (misal 1 km x 1 km) yang kemungkinan terbakar pada saat satelit melintas pada kondisi relatif bebas awan dengan menggunakan algoritme tertentu (Giglio, 2003).

*Hotspot* adalah titik panas yang memiliki tanda-tanda lokasi kebakaran hutan. Parameter ini telah digunakan secara luas di banyak Negara untuk pemantauan kebakaran hutan dan lahan oleh satelit. Deteksi kebakaran hutan dan lahan dilakukan dengan cara pengamatan titik panas (*hotspot*). Titik panas (*hotspot*) dideteksi dengan satelit NOAA (*National Oceanic and Atmospheric Administration*) yang dilengkapi sensor AVHRR (*Advanced Very High Resolution Radiometer*). Dalam mendeteksi kebakaran hutan, satelit NOAA tidak mendeteksi kebakaran (suhu) secara langsung tetapi yang dideteksi adalah *hotspot* (Heryalianto, 2006).

Data titik panas bumi di dapat dari *The earth Observing Data and information System* (EOSDIS), EOSDIS adalah program NASA untuk meneliti bumi. EOSDIS menyediakan kemampuan end-to-end untuk mengelola data ilmu bumi NASA dari berbagai sumber seperti satelit, pesawat, pengukuran lapangan, dan berbagai program lainnya. Untuk misi satelit EOS, EOSDIS menyediakan kemampuan untuk komando dan kontrol, penjadwalan, pengambilan data(earthdata.nasa.gov).

Parameter yang di gunakan dalam penelitian ini adalah *brightness* dan FRP. Akan dijelaskan sebagai berikut:

1. *Brightness* adalah kecerahan piksel api diukur dalam Kelvin.
2. FRP (*fire radiative power*) menggambarkan piksel kekuatan radiasi api dalam MW (megawatt) diambil dari deteksi satelit NOAA.
3. *Confidence* tingkat kepercayaan titik panas dengan *range* 0-100%.

## 2.6 Silhouette Coefficient

Metode evaluasi dalam penelitian ini menggunakan *silhouette Coefision* mengacu pada metode penafsiran dan validasi konsistensi dalam *cluster* data. Teknik ini merepresentasi grafis singkat dari seberapa baik setiap objek yang ada didalam suatu *cluster*. *silhouette* nya adalah ukuran dari seberapa mirip sebuah objek untuk *cluster* itu sendiri (kohesi) dibandingkan dengan kelompok lainnya (pemisahan). Nilai *silhouette* berkisar dari -1 sampai 1, dimana nilai tinggi menunjukkan bahwa objek tersebut sangat cocok untuk *cluster* itu sendiri dan kurang cocok untuk *cluster* lainnya. Keuntungan menggunakan *Silhouette* adalah hanya bergantung pada partisi objek, dan bukan pada algoritme *clustering* yang digunakan untuk mendapatkannya. *Silhouette* dapat digunakan untuk memperbaiki hasil analisis klaster atau untuk membandingkan keluaran dari algoritme pengelompokan yang berbeda yang diterapkan pada data yang sama (Rousseeuw, 1987).

$$s(i) = \frac{x(i)-y(i)}{\max \{y(i),x(i)\}} \quad (2.4)$$

Keterangan:

$x(i)$  = perhitungan jarak antar data didalam *cluster* yang sama.

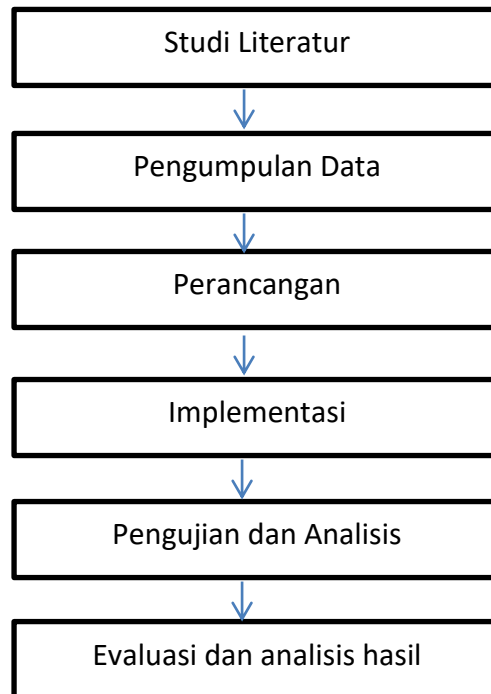
$y(i)$  = perhitungan jarak antar data dengan *cluster* yang berbeda.

## BAB 3 METODOLOGI

Pada bab metodologi ini akan dibahas mengenai metode yang digunakan dan Gambaran penggunaan metode pada implementasi *affinity propagation clustering* panas bumi untuk deteksi titik api kebakaran lahan.

### 3.1 Metodologi Penelitian

Adapun diagram alir pada penelitian ini adalah sebagai berikut:



**Gambar 3. 1 Diagram Alir Metodologi**

Adapun penjelasan secara rinci terkait metodologi yang digunakan pada Gambar 3.1 adalah sebagai berikut:

1. Mempelajari literatur-literatur terkait yang terkait dengan metode *Affinity Propagation*.
2. Mengumpulkan data dan menganalisa data.
3. Melakukan analisis dan perancangan sistem untuk implementasi *affinity propagation clustering* panas bumi untuk deteksi titik api kebakaran lahan.
4. Mengimplementasi hasil perancangan kedalam proses *clustering* yang terkomputerisasi.
5. Melakukan pengujian dan analisis pada aplikasi yang telah dibuat.
6. Mengevaluasi dan menganalisis tingkat kualitas *cluster* berdasarkan perhitungan manualisasi dan hasil implementasi.

### 3.2 Studi Literatur

Untuk menyelesaikan masalah pada penelitian ini agar sesuai dengan tujuan yang akan dicapai maka dibutuhkan studi literatur. adapun pembelajaran teori-teori yang digunakan pada penelitian contohnya pengolahan data mentah yang belum siap digunakan menjadi data yang siap digunakan. Kemudian mempelajari metode *Afiintiy Propagation* sebagai dasar penelitian yang didapat dari jurnal, buku, dan internet serta beberapa literatur lain.

### 3.3 Pengumpulan Data

Data panas bumi di dapat dari *The earth Observing Data and information System* (EOSDIS) NASA (<https://earthdata.nasa.gov/earth-observation-data/near-real-time/firms/active-fire-data#ed-firms-shapefile>). Data yang diambil adalah data dengan fitur *brightness*, *confidence* dan FRP.

### 3.4 Analisis dan Perancangan

Analisis kebutuhan dilakukan dengan cara menentukan kebutuhan apa saja yang dibutuhkan untuk membangun perangkat lunak antarlain deskripsi, perancangan, manualisasi, desain *Interface*, dan perancangan pengujian dan evaluasi. Adapun kebutuhan perangkat keras yang digunakan agar proses implementasi tidak mengalami masalah yaitu memiliki spesifikasi sebagai berikut:

- Sistem operasi Windows 10.
- RAM : 4.00 GB
- Processor Intel(R) Core™ i5-2450M CPU @ 2.50GHz with Turbo Boost up to 3.10 GHz
- VGA card AMD Radeon HD 7670M / Intel HD Graphics 3000
- Aplikasi di bangun dengan menggunakan bahasa pemrograman PHP.

Tahap perancangan adalah tahap pembuatan alur kerja untuk memecahkan permasalahan pengelompokan data menggunakan metode *Affinity Propagation*, penggunaan data kedalam algoritme, desain aplikasi yang menampilkan proses dan hasil akurasi serta desain pengujian.

### 3.5 Implementasi

Implementasi adalah tahap membangun aplikasi yang mengacu pada perancangan dan menerapkan hal yang telah didapat dalam proses studi literatur. Fase –fase yang ada dalam implementasi antara lain:

- Implementasi antarmuka.



- Implementasi basis data, menggunakan sebuah server localhost (XAMPP) yang bertujuan untuk memudahkan melakukan manipulasi dan penyimpanan data.
- Implementasi algoritme, melakukan perhitungan dengan metode studi literatur kedalam bahasa pemrograman php.

### **3.6 Pengujian dan Analisis**

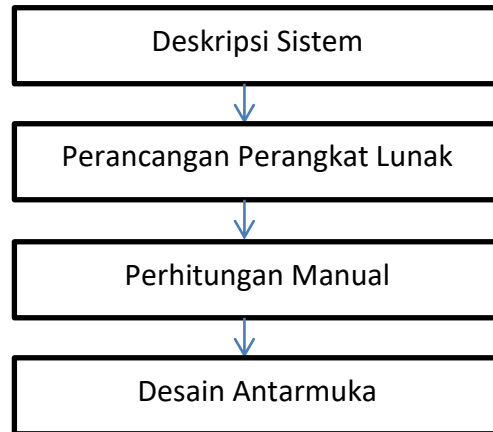
Uji coba sistem dilakukan untuk mengetahui apakah aplikasi berjalan dengan baik dan sesuai dengan perancangan. Hasil dari uji coba tersebut akan dijadikan bahan evaluasi supaya dapat sesuai dengan tujuan yang diharapkan. Adapun pengujian yang digunakan yaitu dengan menggunakan metode *silhouette coefficient* untuk mengukur kesesuaian hasil *clustering*.

### **3.7 Evaluasi dan Analisis Hasil**

Pada tahapan ini akan dilakukan evaluasi untuk mempertimbangkan tingkat keberhasilan implementasi metode pada aplikasi dalam menyelesaikan permasalahan apakah sudah sesuai dengan tujuan penelitian. Kemudian dilanjutkan proses penarikan kesimpulan yang merupakan tahap paling akhir dimana berisikan hasil yang diperoleh dari proses pengujian dan analisis yang telah melalui proses evaluasi. Kemudian dilanjutkan dengan kesimpulan dan saran yang dapat digunakan oleh pembaca sebagai acuan memperbaiki dan mengembangkan aplikasi kedepannya.

## BAB 4 PERANCANGAN

Pada tahap perancangan ini terdiri dari deskripsi sistem secara umum, perancangan aplikasi, manualisasi, dan desain antarmuka. Pada Gambar 4.1 di bawah ini merupakan model perancangan yang digunakan pada penelitian ini.



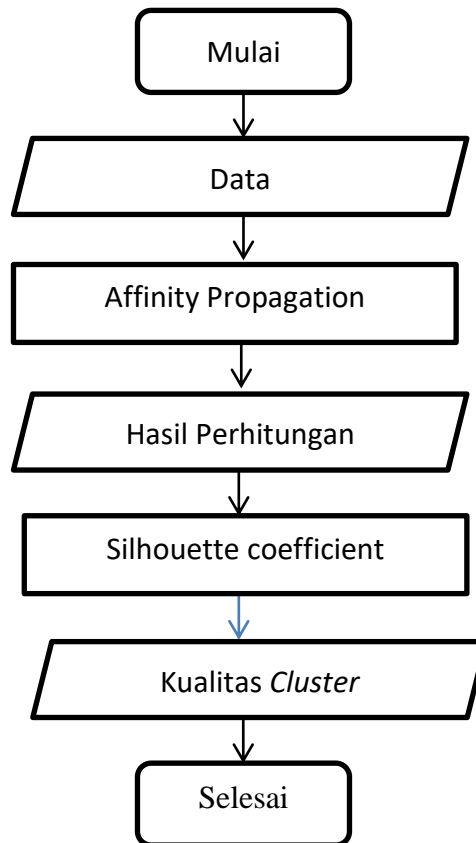
Gambar 4. 1 Diagram Alir Perancangan Sistem

### 4.1 Deskripsi Sistem

Penelitian ini dilakukan dengan menerapkan metode *Affinity Propagation* dalam penentuan *cluster* titik api. dimana metode *Affinity Propagation* akan melakukan *clustering* pada data, terdapat data dari EOSDIS NASA yang digunakan di penelitian ini. Sedangkan untuk *input* ada 2 parameter yaitu *brightness* dan FRP. *Output* dari aplikasi ini yaitu *cluster* panas bumi.

### 4.2 Perancangan Perangkat Lunak

Tahap perancangan perangkat lunak pada proses implementasi *affinity propagation clustering* panas bumi yang dijalankan pada program aplikasi diGambarkan pada diagram alir sebelumnya. Pada prosesnya terdapat parameter yang digunakan, adapun parameter yang digunakan adalah *brightness* dan FRP. Adapun diagram alur dari jalanya proses perhitungan ditunjukkan pada Gambar 4.2.

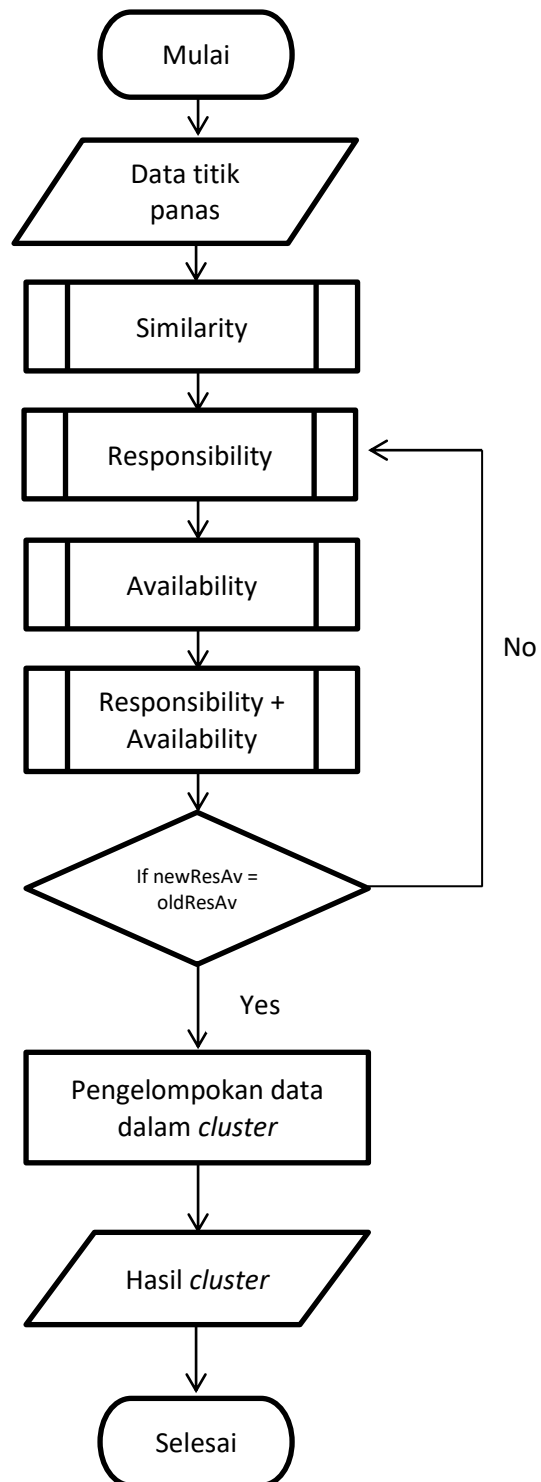


Gambar 4. 2 Diagram Alur Perancangan

#### 4.2.1 Affinity Propagation

Proses *clustering Affinity Propagation* dimulai dengan mencari nilai kesamaan tiap data atau *Similarity*. Dilanjutkan menentukan *responsibility* menggunakan hasil dari proses *similarity*, setelah ditemukan nilai *responsibility* dilanjutkan mencari nilai *availability*. Dalam melakukan proses *responsibility* dan *availability* dilakukan iterasi sampai *konvergen*, dimana nilai *availability* tidak berubah dalam iterasi selanjutnya. Kemudian dilakukan penjumlahan nilai *responsibility* dan *availability* untuk menentukan *cluster*.

*Flowchart* diagram untuk *Affinity Propagation* ditunjukan pada Gambar 4.3.



Gambar 4. 3 flowchart affinity propagation

### 4.3 Perhitungan Manual

Perhitungan manual disertakan untuk membantu menjelaskan tahapan perancangan secara sistematis. Perhitungan manual dilakukan dengan ketentuan sebagai berikut:

#### 4.3.1 Affinity Propagation

1. Data yang digunakan yaitu 10 data sampel dari *The earth Observing Data and information System* (EOSDIS) NASA dengan 2 parameter yaitu *brightness* dan FRP.
2. Tahapan yang digunakan terdiri dari pencarian *similarity* atau negatif *Euclidean distance* dari tiap titik, menentukan *responsibility*, menentukan *availability*, dan menentukan *cluster*.

Data yang digunakan dalam Perhitungan manual ditampilkan pada Tabel 4.1 terdapat 5 parameter dan 10 data titik panas bumi.

**Tabel 4. 1 data panas bumi**

TITIK	BRIGHTNESS	FRP
A	327.6	35.1
B	319.3	18.7
C	323.2	24.4
D	321.5	17.7
E	325.6	24.3
F	348	73.6
G	321.8	44.4
H	324.6	18.7
I	333.7	74.7
J	336.1	65.4

#### Langkah 1

Tahapan ini digunakan untuk mencari *similarity* titik dengan titik yang lain dengan cara *negative euclidean distance*. Perhitungan ini merupakan contoh merupakan contoh perhitungan *similarity*  $S(A,B)$  sampai  $S(A,C)$ .

$$\begin{aligned} S(A,B) &= -\sqrt{(327.6 - 319.3)^2 + (35.1 - 18.7)^2} \\ &= -18.3806964 \end{aligned}$$

$$\begin{aligned} S(A,C) &= -\sqrt{(327.6 - 323.2)^2 + (35.1 - 24.4)^2} \\ &= -11.56935608 \end{aligned}$$

Hasil keseluruhan Perhitungan bisa dilihat pada Tabel 4.2.

**Tabel 4. 2 perhitungan *Similarity* tiap titik data**

A -> B	-18.3806964	A -> C	-11.56935608	A -> D	-18.43827541
B -> C	-6.90651866	B -> D	-2.416609195	B -> E	-8.429116205
C -> D	-6.912307864	C -> E	-2.40208243	C -> F	-55.09700536
D -> E	-7.769813383	D -> F	-61.86323626	D -> G	-26.70168534
E -> F	-54.15025392	E -> G	-20.45605045	E -> H	-5.688585061
F -> G	-39.2311101	F -> H	-59.67889074	F -> I	-14.34224529
G -> H	-25.85207922	G -> I	-32.55303365	G -> J	-25.40649523
H -> I	-56.73455737	H -> J	-48.0951141	A -> J	-31.46966794
I -> J	-9.604686356	C -> J	-42.98150765	B -> I	-57.8217952
A -> E	-10.98362417	A -> F	-43.57074707	A -> H	-16.67213244
B -> F	-61.94917271	B -> G	-25.82130903	D -> J	-49.88436629
C -> G	-20.04894012	C -> H	-5.869412236	A -> I	-40.06706877
D -> H	-3.257299495	D -> I	-58.29099416	B -> J	-49.62993049
E -> I	-51.04674328	E -> J	-42.42004243	C -> I	-51.38423883
F -> J	-14.45164351	A -> G	-10.9603832	B -> H	-5.3

## Langkah 2

Berdasarkan Perhitungan pada Tabel 4.2 akan di cari *responsibility* dengan cara memasukan nilai *similarity* tiap titik dikurangi nilai maksimal dari *availability* (a) yang di asumsikan 0 untuk iterasi pertama, ditambah *similarity* selain titik yang dikurangkan sesuai dengan persamaan dan contoh perhitungan berikut:

$$r(i, k) \leftarrow s(i, k) - \max_{k' \neq k} \{a(i, k') + s(i, k')\}$$

$$\begin{aligned}
 r(A, B) &= s(A, B) - \max \{ (a(A, C) + s(A, C)), (a(A, D) + s(A, D)), (a(A, E) + s(A, E)), (a(A, F) + s(A, F)), \\
 &\quad (a(A, G) + s(A, G)), (a(A, H) + s(A, H)), (a(A, I) + s(A, I)), (a(A, J) + s(A, J)) \} \\
 &= -18.3806964 - \max((0 + (-11.5693)), (0 + (-18.4382)), (0 + (-10.9836)), (0 + (-43.5707)), (0 + (-10.9603)), (0 + (-16.6721)), (0 + (-40.0670)), (0 + (-31.4696))) \\
 &= -7.420313191.
 \end{aligned}$$

### Menghitung *self-responsibility*

$$r(k,k) = s(k,k) - \max(s(l,k))$$

$$r(A,A) = s(A,A) - \max(s(A,B), s(A,C), s(A,D), s(A,E), s(A,F), s(A,G), s(A,H), s(A,I), s(A,J))$$

$$r(A,A) = 0 - \max(-18.3806964, -18.4382, -10.9836, -43.5707, -10.9603, -16.6721, 40.0670, -31.4696) \\ = 10.9603832.$$

Hasil keseluruhan perhitungan langkah 2 dapat dilihat pada Tabel 4.3 Perhitungan *responsibility* tiap titik data dan pada Tabel 4.4 perhitungan *self-responsibility* tiap titik data.

**Tabel 4. 3 Perhitungan *responsibility* tiap titik data**

TITIK A		TITIK B		TITIK C		TITIK D	
r(A,B)	-7.420313191	r(B,A)	-15.9640872	r(C,A)	-9.167273646	r(D,A)	-16.02166621
r(A,C)	-0.60897287	r(B,C)	-4.489909465	r(C,B)	-4.50443623	r(D,B)	0.8406903
r(A,D)	-7.477892202	r(B,D)	2.883390805	r(C,D)	-4.510225434	r(D,C)	-4.495698669
r(A,E)	-0.023240969	r(B,E)	-6.01250701	r(C,E)	3.467329806	r(D,E)	-5.353204188
r(A,F)	-32.61036386	r(B,F)	-59.53256352	r(C,F)	-52.69492293	r(D,F)	-59.44662706
r(A,G)	0.023240969	r(B,G)	-23.40469983	r(C,G)	-17.64685769	r(D,G)	-24.28507615
r(A,H)	-5.711749232	r(B,H)	-2.883390805	r(C,H)	-3.467329806	r(D,H)	-0.8406903
r(A,I)	-29.10668557	r(B,I)	-55.405186	r(C,I)	-48.9821564	r(D,I)	-55.87438496
r(A,J)	-20.50928473	r(B,J)	-47.21332129	r(C,J)	-40.57942522	r(D,J)	-47.46775709
TITIK E		TITIK F		TITIK G		TITIK H	
r(E,A)	-8.581541744	r(F,A)	-29.22850178	r(G,A)	9.088556917	r(H,A)	-13.41483294
r(E,B)	-6.027033775	r(F,B)	-47.60692742	r(G,B)	-14.86092582	r(H,B)	-2.042700505
r(E,C)	3.286502631	r(F,C)	-40.75476007	r(G,C)	-9.088556917	r(H,C)	-2.612112741
r(E,D)	-5.367730953	r(F,D)	-47.52099096	r(G,D)	-15.74130214	r(H,D)	2.042700505
r(E,F)	-51.74817149	r(F,E)	-39.80800863	r(G,E)	-9.495667245	r(H,E)	-2.431285566
r(E,G)	-18.05396802	r(F,G)	-24.88886481	r(G,F)	-28.2707269	r(H,F)	-56.42159125
r(E,H)	-3.286502631	r(F,H)	-45.33664545	r(G,H)	-14.89169602	r(H,G)	-22.59477973
r(E,I)	-48.64466086	r(F,I)	0.109398214	r(G,I)	-21.59265045	r(H,I)	-53.47725788
r(E,J)	-40.01796	r(F,J)	-0.109398214	r(G,J)	-14.44611203	r(H,J)	-44.8378146

TITIK I		TITIK J	
r(I,A)	-30.4623824	r(J,A)	-21.8649816
r(I,B)	-48.2171088	r(J,B)	-40.0252441
r(I,C)	-41.7795525	r(J,C)	-33.3768213
r(I,D)	-48.6863078	r(J,D)	-40.2796799
r(I,E)	-41.4420569	r(J,E)	-32.8153561
r(I,F)	-4.73755894	r(J,F)	-4.84695715
r(I,G)	-22.9483473	r(J,G)	-15.8018089
r(I,H)	-47.129871	r(J,H)	-38.4904277
r(I,J)	4.737558935	r(J,I)	4.846957149

**Tabel 4. 4** perhitungan *self-responsibility* tiap titik data

self-responsibility	
r(A,A)	10.9603832
r(B,B)	2.416609195
r(C,C)	2.40208243
r(D,D)	2.416609195
r(E,E)	2.40208243
r(F,F)	14.34224529
r(G,G)	10.9603832
r(H,H)	3.257299495
r(I,I)	9.604686356
r(J,J)	9.604686356

### Langkah 3

Berdasarkan Perhitungan pada Tabel 4.3 akan di cari *availability* dengan cara minimal dari 0 dan *self-responsibility* ditambah nilai maksimal dari  $(0, r(i', k))$  sesuai dengan persamaan dan contoh perhitungan berikut:



$$a(i, k) \leftarrow \min \{0, r, (k, k) + \sum_{i' \notin \{i, k\}} \max(0, r(i', k))\}$$

$$\begin{aligned} a(A, B) &= \min \{0, r(B, B) + \max (0, r(B, B), r(C, B), r(D, B), r(E, B), r(F, B), r(G, B), r(H, B), \\ &\quad r(I, B), r(J, B))\} \\ &= \min \{0, 2.416609195 + \max (0, 2.416609195, -4.50443623, 0.8406903, - \\ &\quad 6.027033775, -47.60692742, -14.86092582, -2.042700505, - \\ &\quad 48.21710884, -40.02524413)\} \\ &= 4.833218389. \end{aligned}$$

Menghitung *self-availability*.

$$a(k, k) \leftarrow \sum_{i' \neq k} \max(0, r(i', k))$$

$$\begin{aligned} a(A, A) &= \max (0, r(A, A), r(B, A), r(C, A), r(D, A), r(E, A), r(F, A), r(G, A), r(H, A), r(I, A), \\ &\quad r(J, A)) \\ &= \max (0, 10.9603832, -15.9640872, -9.167273646, -16.02166621, - \\ &\quad 8.581541744, -29.22850178, 9.088556917, -13.41483294, -30.46238242, \\ &\quad -21.86498158) \\ &= 10.9603832. \end{aligned}$$

Hasil keseluruhan perhitungan langkah 3 dapat dilihat pada Tabel 4.5 perhitungan *availability* tiap titik data dan pada Tabel 4.6 perhitungan *self-availability* tiap titik data.

**Tabel 4. 5 perhitungan *availability* tiap titik data.**

TITIK A		TITIK B		TITIK C		TITIK D	
a(A,B)	4.833218389	a(B,A)	21.92076641	a(C,A)	21.92076641	a(D,A)	21.92076641
a(A,C)	5.688585061	a(B,C)	5.688585061	a(C,B)	4.833218389	a(D,B)	4.833218389
a(A,D)	5.3	a(B,D)	4.833218389	a(C,D)	5.3	a(D,C)	5.688585061
a(A,E)	5.869412236	a(B,E)	5.869412236	a(C,E)	4.80416486	a(D,E)	5.869412236
a(A,F)	28.68449058	a(B,F)	28.68449058	a(C,F)	28.68449058	a(D,F)	28.68449058
a(A,G)	21.92076641	a(B,G)	21.92076641	a(C,G)	21.92076641	a(D,G)	21.92076641
a(A,H)	6.51459899	a(B,H)	6.51459899	a(C,H)	6.51459899	a(D,H)	6.51459899
a(A,I)	19.20937271	a(B,I)	19.20937271	a(C,I)	19.20937271	a(D,I)	19.20937271
a(A,J)	19.20937271	a(B,J)	19.20937271	a(C,J)	19.20937271	a(D,J)	19.20937271

TITIK E		TITIK F		TITIK G		TITIK H	
a(E,A)	21.92076641	a(F,A)	21.92076641	a(G,A)	21.92076641	a(H,A)	21.92076641
a(E,B)	4.833218389	a(F,B)	4.833218389	a(G,B)	4.833218389	a(H,B)	4.833218389
a(E,C)	4.80416486	a(F,C)	5.688585061	a(G,C)	5.688585061	a(H,C)	5.688585061
a(E,D)	5.3	a(F,D)	5.3	a(G,D)	5.3	a(H,D)	5.3
a(E,F)	28.68449058	a(F,E)	5.869412236	a(G,E)	5.869412236	a(H,E)	5.869412236
a(E,G)	21.92076641	a(F,G)	21.92076641	a(G,F)	28.68449058	a(H,F)	28.68449058
a(E,H)	6.51459899	a(F,H)	6.51459899	a(G,H)	6.51459899	a(H,G)	21.92076641
a(E,I)	19.20937271	a(F,I)	19.20937271	a(G,I)	19.20937271	a(H,I)	19.20937271
a(E,J)	19.20937271	a(F,J)	19.20937271	a(G,J)	19.20937271	a(H,J)	19.20937271
TITIK I		TITIK J					
a(I,A)	21.92076641	a(J,A)	21.92076641				
a(I,B)	4.833218389	a(J,B)	4.833218389				
a(I,C)	5.688585061	a(J,C)	5.688585061				
a(I,D)	5.3	a(J,D)	5.3				
a(I,E)	5.869412236	a(J,E)	5.869412236				
a(I,F)	28.68449058	a(J,F)	28.68449058				
a(I,G)	21.92076641	a(J,G)	21.92076641				
a(I,H)	6.51459899	a(J,H)	6.51459899				
a(I,I)	19.20937271	a(J,I)	19.20937271				

**Tabel 4. 6 perhitungan *self-availability* tiap titik data.**

self-availability	
a(A,A)	10.9603832
a(B,B)	2.416609195
a(C,C)	3.286502631
a(D,D)	2.883390805
a(E,E)	3.467329806

Self-availability	
a(F,F)	14.34224529
a(G,G)	10.9603832
a(H,H)	3.257299495
a(I,I)	4.846957149
a(J,J)	9.604686356

#### Langkah 4

Menjumlahkan *responsibility* dan *availability*. Langkah 1 – 4 diulangi terus menerus sampai nilai pada penjumlahan *responsibility* dan *availability* tidak berubah atau setelah 10 iterasi. Berikut adalah contoh penjumlahan *responsibility* dan *availability* setelah dilakukan 3 iterasi dan nilainya tidak berubah dapat dilihat pada Tabel 4.7.

**Tabel 4. 7 perhitungan responsibility + availability**

TITIK A		TITIK B		TITIK C		TITIK D	
a+r(A,B)	-24.50786121	a+r(B,A)	1.123460819	a+r(C,A)	7.949327905	a+r(D,A)	0.225191508
a+r(A,C)	-17.72557442	a+r(B,C)	-5.642423814	a+r(C,B)	-12.42471061	a+r(D,B)	-1.065881808
a+r(A,D)	-24.56544022	a+r(B,D)	-1.123460819	a+r(C,D)	-12.43049981	a+r(D,C)	-5.590634006
a+r(A,E)	-17.13984252	a+r(B,E)	-7.165021359	a+r(C,E)	-7.949327905	a+r(D,E)	-6.448139525
a+r(A,F)	-25.84663969	a+r(B,F)	-36.80475215	a+r(C,F)	-36.76392511	a+r(D,F)	-36.66123668
a+r(A,G)	0	a+r(B,G)	-7.440612634	a+r(C,G)	-8.479584046	a+r(D,G)	-8.263409933
a+r(A,H)	-21.11791665	a+r(B,H)	-2.325471024	a+r(C,H)	-9.706223581	a+r(D,H)	-0.225191508
a+r(A,I)	-31.81807926	a+r(B,I)	-42.1524925	a+r(C,I)	-42.52627645	a+r(D,I)	-42.56411245
a+r(A,J)	-23.22067843	a+r(B,J)	-33.96062779	a+r(C,J)	-34.12354527	a+r(D,J)	-34.15748458
TITIK E		TITIK F		TITIK G		TITIK H	
a+r(E,A)	8.535059806	a+r(F,A)	-35.99222595	a+r(G,A)	0	a+r(H,A)	1.991334478
a+r(E,B)	-14.53304005	a+r(F,B)	-71.45819962	a+r(G,B)	-31.94847385	a+r(H,B)	-5.715415583
a+r(E,C)	-8.535059806	a+r(F,C)	-64.63508579	a+r(G,C)	-26.20515847	a+r(H,C)	-6.313881349
a+r(E,D)	-13.87373723	a+r(F,D)	-71.37226316	a+r(G,D)	-32.82885016	a+r(H,D)	-3.672715078
a+r(E,F)	-36.40290558	a+r(F,E)	-63.68833436	a+r(G,E)	-26.61226879	a+r(H,E)	-6.133054174

TITIK E		TITIK F		TITIK G		TITIK H	
a+r(E,G)	-9.472426275	a+r(F,G)	-31.65258899	a+r(G,F)	-21.50700273	a+r(H,F)	-36.24303413
a+r(E,H)	-10.11112831	a+r(F,H)	-67.50653704	a+r(G,H)	-30.29786344	a+r(H,G)	-9.179946783
a+r(E,I)	-42.77451281	a+r(F,I)	-9.475117871	a+r(G,I)	-24.30404415	a+r(H,I)	-42.77381863
a+r(E,J)	-34.14781196	a+r(F,J)	-9.584516084	a+r(G,J)	-17.15750573	a+r(H,J)	-34.13437536
TITIK I		TITIK J					
a+r(I,A)	-32.4885477	a+r(J,A)	-23.7817486				
a+r(I,B)	-67.3308221	a+r(J,B)	-59.0295592				
a+r(I,C)	-60.9223193	a+r(J,C)	-52.4101899				
a+r(I,D)	-67.8000211	a+r(J,D)	-59.283995				
a+r(I,E)	-60.5848237	a+r(J,E)	-51.8487247				
a+r(I,F)	4.737558935	a+r(J,F)	4.628160722				
a+r(I,G)	-24.9745125	a+r(J,G)	-17.7185759				
a+r(I,H)	-64.5622037	a+r(J,H)	-55.8133622				
a+r(I,J)	-4.73755894	a+r(J,I)	-4.62816072				

perhitungan *responsibility* + *availability* dan dari perhitungan diatas ditemukan titik dengan maksimal *responsibility* + *availability* sebagai eksemplar yaitu A dan F terbentuk 2 *cluster*. pembentukan *cluster* dapat dilihat pada Tabel 4.8 pembentukan *cluster*.

**Tabel 4. 8 pembentukan *cluster***

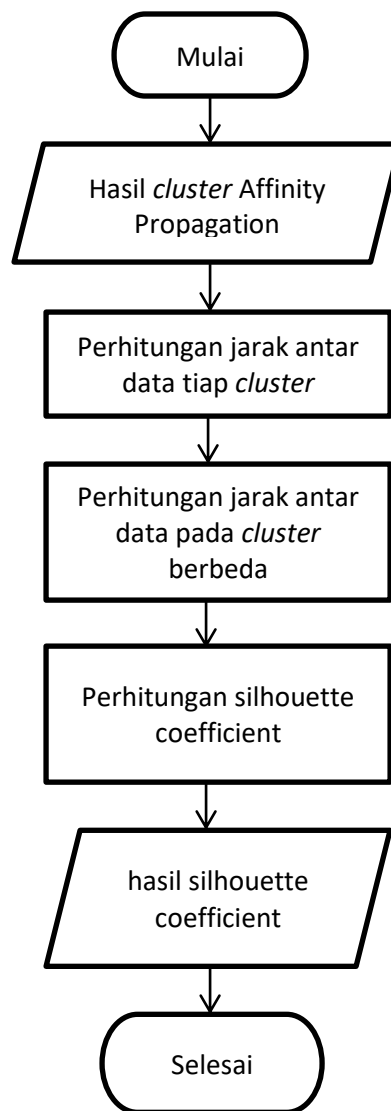
CLUSTER 1			CLUSTER 2		
TITIK	BRIGHTNESS	FRP	TITIK	BRIGHTNESS	FRP
A	327.6	35.1	F	348	73.6
B	319.3	18.7	I	333.7	74.7
C	323.2	24.4	J	336.1	65.4
D	321.5	17.7			
E	325.6	24.3			
G	321.8	44.4			
H	324.6	18.7			

#### 4.3.2 Silhouette Coefficient

Setelah proses perhitungan *Affinity Propagation* selesai, maka akan dilakukan perhitungan *Silhouette Coefficient*. *Silhouette Coefficient* digunakan untuk

menilai kekuatan dan kualitas *cluster* dan seberapa baik suatu data ditempatkan dalam suatu *cluster*.

*Flowchart* diagram untuk *silhouette coefficient* ditunjukkan pada Gambar 4.4.



**Gambar 4. 4 Flowchart Silhouette Coefficient**

Pembentukan *Cluster* sudah dilakukan pada sub bab perhitungan *Affinity Propagation* yang dapat dilihat pada Tabel 4.8 yang selanjutnya data tersebut akan diolah dengan *Silhouette Coefficient* sebagai berikut:

#### **Langkah 1**

Mencari  $x(i)$  dengan cara menghitung rata-rata jarak titik  $i$  dengan semua titik yang ada dalam satu *cluster* yang sama. Contoh Jarak titik A dengan titik B menggunakan:

$$\text{Jarak} = \sqrt{(327.6 - 319.3)^2 + (35.1 - 18.7)^2} = 18.3807$$

Jika jarak titik satu dengan semua titik dalam *cluster* sama sudah ditemukan maka bisa dihitung  $x(i)$  sebagai berikut:

$$\begin{aligned} x(i) &= \frac{0+18.3807+\dots+10.9603+16.6721}{7} \\ &= 12.42920967 \end{aligned}$$

Keseluruhan hasil  $x(i)$  dapat dilihat pada Tabel 4.9.

**Tabel 4. 9 hasil  $x(i)$**

CLUSTER 1		CLUSTER 2	
TITIK	NILAI	TITIK	NILAI
A	12.42920967	F	9.597962932
B	9.607749926	I	7.982310549
C	7.672659627	J	8.01877662
D	9.356570098		
E	7.961324529		
G	16.16919956		
H	5.655347033		

## Langkah 2

Mencari  $y(i)$  dengan cara menghitung rata-rata jarak titik  $i$  dengan semua titik yang ada pada *cluster* berbeda. Contoh jarak titik A dengan titik F menggunakan:

$$\text{Jarak} = \sqrt{(327.6 - 348)^2 + (35.1 - 73.6)^2} = 43.57075$$

Jika jarak titik satu dengan semua titik dalam *cluster* yang berbeda sudah ditemukan maka bisa dihitung  $y(i)$  sebagai berikut:

$$b(i) = \frac{0+43.57075+40.06707+31.46967}{4} = 28.77687094$$

Keseluruhan hasil  $y(i)$  dapat dilihat pada Tabel 4.10.

**Tabel 4. 10 hasil  $y(i)$**

CLUSTER 1		CLUSTER 2	
TITIK	NILAI	TITIK	NILAI
A	28.77687094	F	46.94255202
B	42.3502246	I	43.48730391
C	37.36568796	J	36.23589052
D	42.50964918		
E	36.90425991		
G	24.29765975		
H	41.12714055		

### Langkah 3

Setelah  $x(i)$  dan  $y(i)$  diketahui maka selanjutnya akan di cari nilai *silhouette coefficient* atau  $s(i)$  menggunakan rumus pada persamaan 2.4.

Contoh mencari nilai  $s(i)$  pada titik A:

$$s(A) = \frac{12.42920967 - 28.77687094}{\max(12.42920967, 28.77687094)} = 0.568083351$$

Nilai keseluruhan  $s(i)$  tiap data dapat dilihat pada Tabel 4.11

**Tabel 4. 11 hasil  $s(i)$**

A	0.568083351	F	0.795538109
B	0.773135798	I	0.816445035
C	0.794660287	J	0.778706236
D	0.779895382		
E	0.784270853		
G	0.334536753		
H	0.86249112		

Selannjutnya dicari nilai akhir  $s(i)$  dengan cara menghitung nilai rata-rata  $s(i)$ .

$$S_i = \frac{0.568+0.7731+0.7946+0.7798+0.7842+0.3345+0.8624+0.7955+0.8164+0.7787}{10} \\ = 0.728776292$$

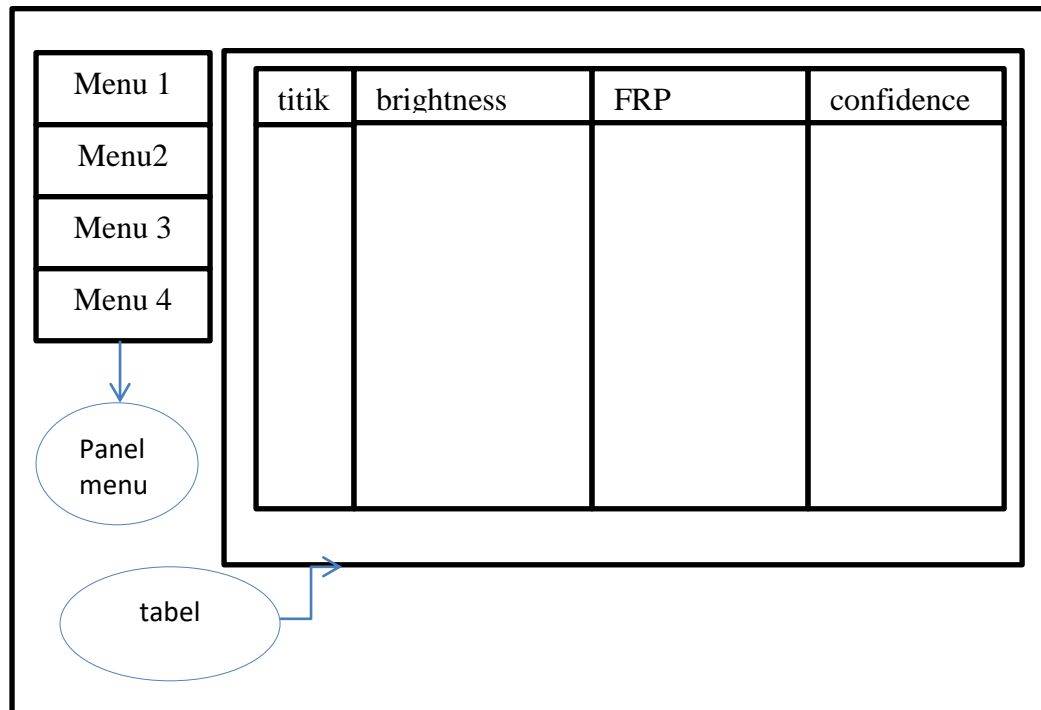
Dengan demikian telah didapat nilai *Silhouette Coefficient* hasil *clustering Affinity Propagation* adalah 0.728776292.

## 4.4 Desain Antarmuka Aplikasi

Pada bagian ini akan ditunjukkan beberapa rancangan antarmuka (*interface*) pada aplikasi ini.

### 4.4.1 Perancangan Antarmuka Data

Perancangan antarmuka pada Gambar 4.5 Ditujukan untuk menampilkan dataset yang akan digunakan pada metode *Affinity Propagation*.



**Gambar 4. 5 Rancang Antarmuka Data**

Penjelasan rancangan antarmuka data diatas ditunjukkan pada Tabel 4.12.

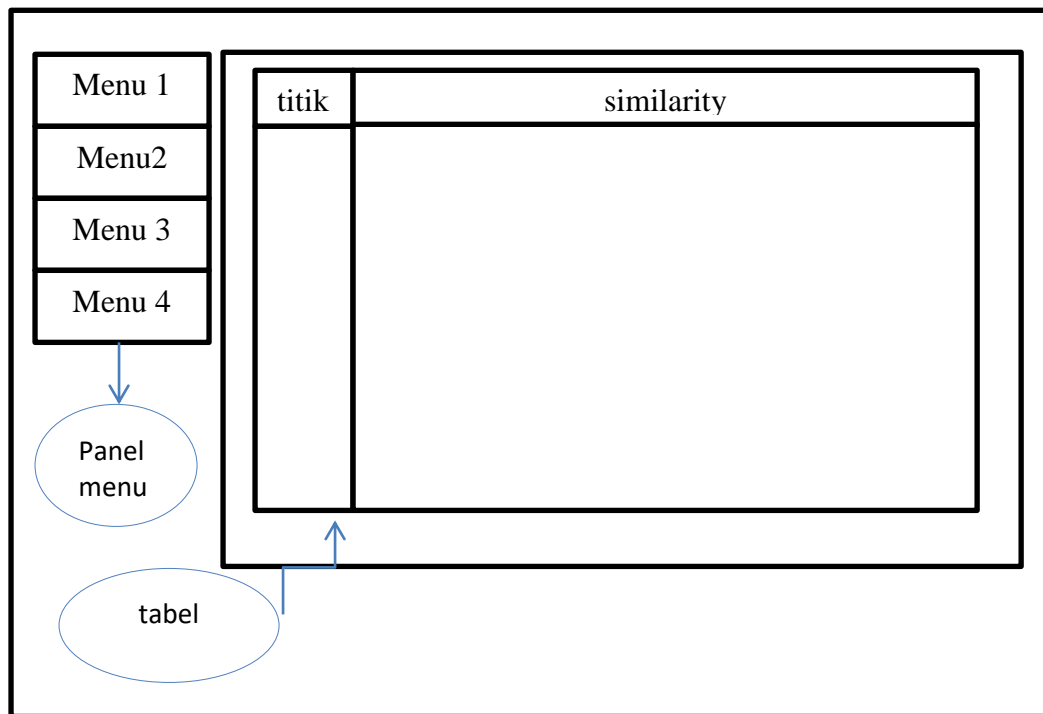
**Tabel 4. 12 Penjelasan Antarmuka Data**

nama	penjelasan
Panel menu	Terdapat berbagai pilihan menu pada aplikasi dan saling terhubung untuk melihat proses aplikasi.
Tabel	Digunakan untuk manampilkan semua data yang akan diolah pada aplikasi.



#### 4.4.2 Perancangan Antarmuka Similarity

Perancangan antarmuka pada Gambar 4.6 ditujukan untuk menampilkan hasil dari proses *Similarity* pada metode *Affinity Propagation*.



**Gambar 4. 6 Rancang Antarmuka Similarity**

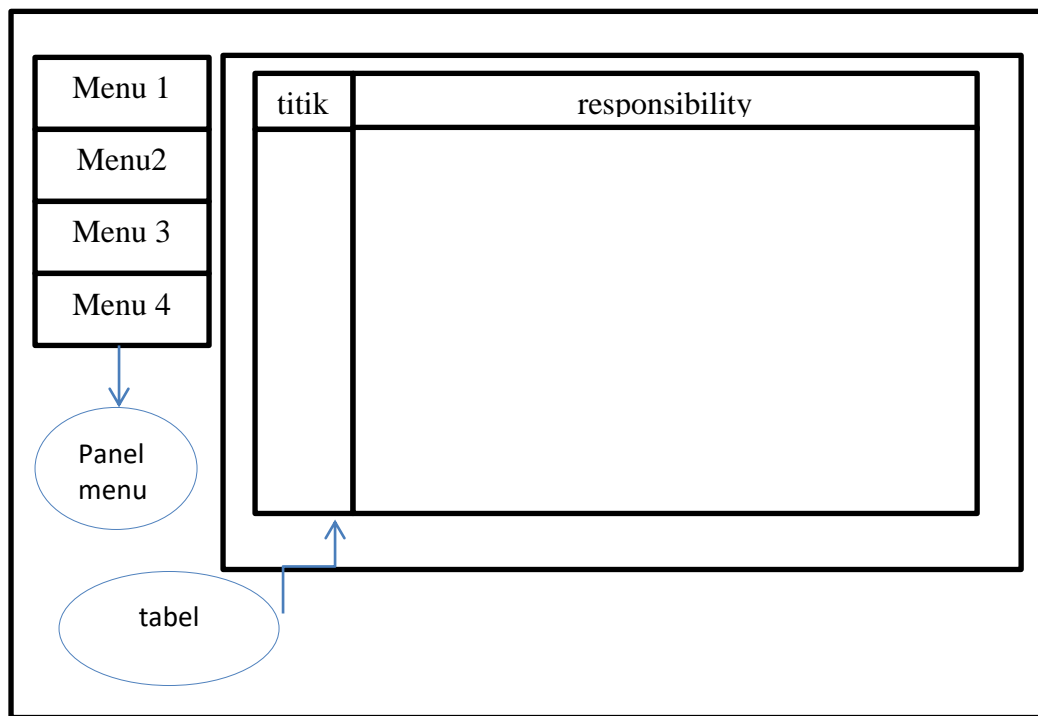
Penjelasan rancangan antarmuka *similarity* diatas ditunjukkan pada Tabel 4.13:

**Tabel 4. 13 Penjelasan Rancang Antarmuka Similarity**

nama	penjelasan
Panel menu	Terdapat berbagai pilihan menu pada aplikasi dan saling terhubung untuk melihat proses aplikasi.
Tabel	Digunakan untuk manampilkan semua hasil <i>similarity</i> tiap titik data yang akan diolah pada aplikasi.

#### 4.4.3 Perancangan Antarmuka Responsibility

Perancangan antarmuka pada Gambar 4.7 ditujukan untuk menampilkan hasil dari proses *Responsibility* pada metode *Affinity Propagation*.



**Gambar 4. 7 Rancang Antarmuka Responsibility**

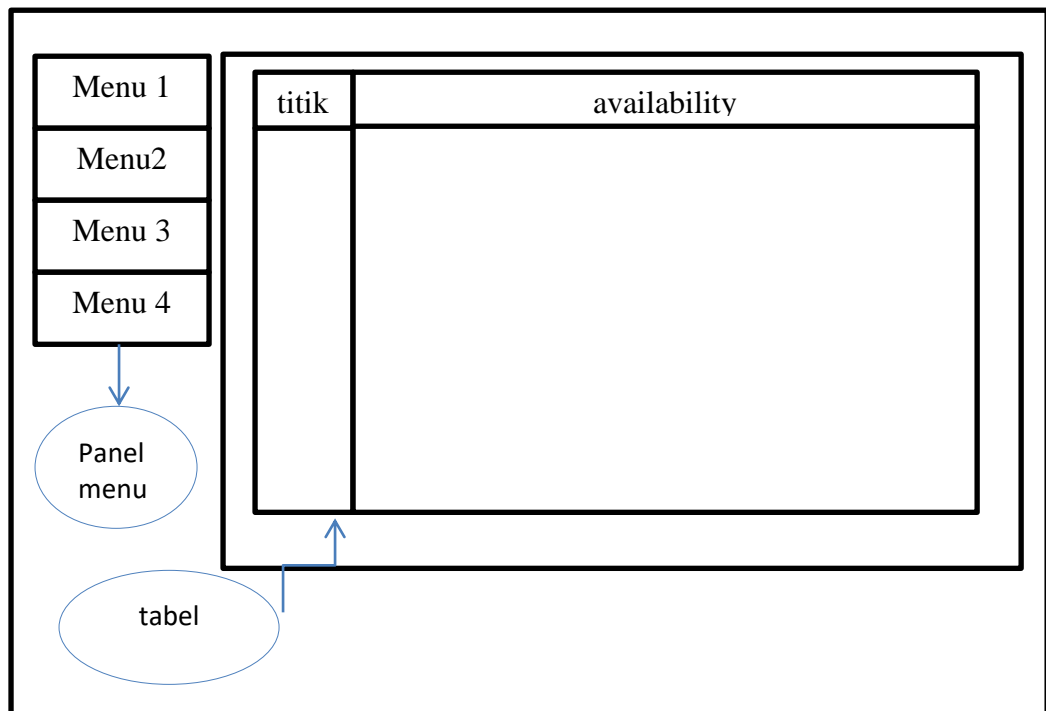
Penjelasan rancangan antarmuka *responsibility* diatas ditunjukan pada Tabel 4.14:

**Tabel 4. 14 Penjelasan Rancang Antarmuka Responsibility**

nama	penjelasan
Panel menu	Terdapat berbagai pilihan menu pada aplikasi dan saling terhubung untuk melihat proses aplikasi.
Tabel	Digunakan untuk manampilkan semua hasil <i>responsibility</i> tiap titik data yang akan diolah pada aplikasi.

#### 4.4.4 Perancangan Antarmuka Availability

Perancangan antarmuka pada Gambar 4.8 ditujukan untuk menampilkan hasil dari proses *Availability* pada metode *Affinity Propagation*.



Gambar 4. 8 Rancang Antarmuka Availability

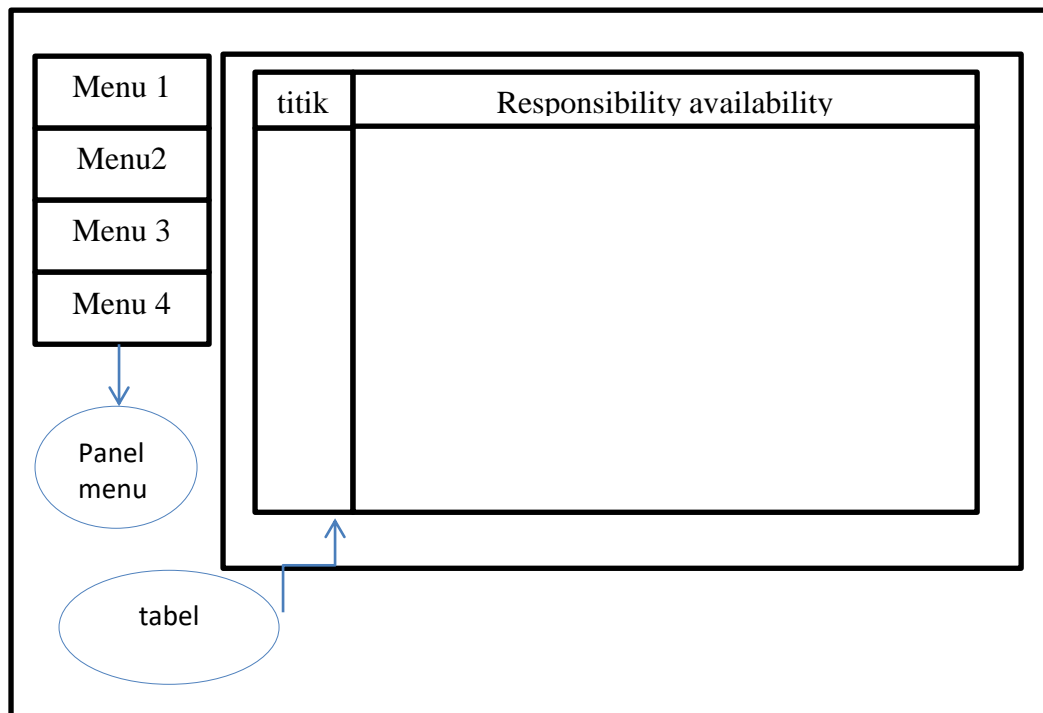
Penjelasan rancangan antarmuka *availability* diatas ditunjukan pada Tabel 4.15.

Tabel 4. 15 Penjelasan Antarmuka Availability

nama	penjelasan
Panel menu	Terdapat berbagai pilihan menu pada aplikasi dan saling terhubung untuk melihat proses aplikasi.
Tabel	Digunakan untuk manampilkan semua hasil <i>availability</i> tiap titik data yang akan diolah pada aplikasi.

#### 4.4.5 Perancangan Antarmuka Sum Responsibility dan Availability

Perancangan antarmuka pada Gambar 4.9 ditujukan untuk menampilkan hasil dari proses penjumlahan *Responsibility* dan *Availability* pada metode *Affinity Propagation*.



**Gambar 4. 9 Rancang Antarmuka Sum Responsibility dan Availability**

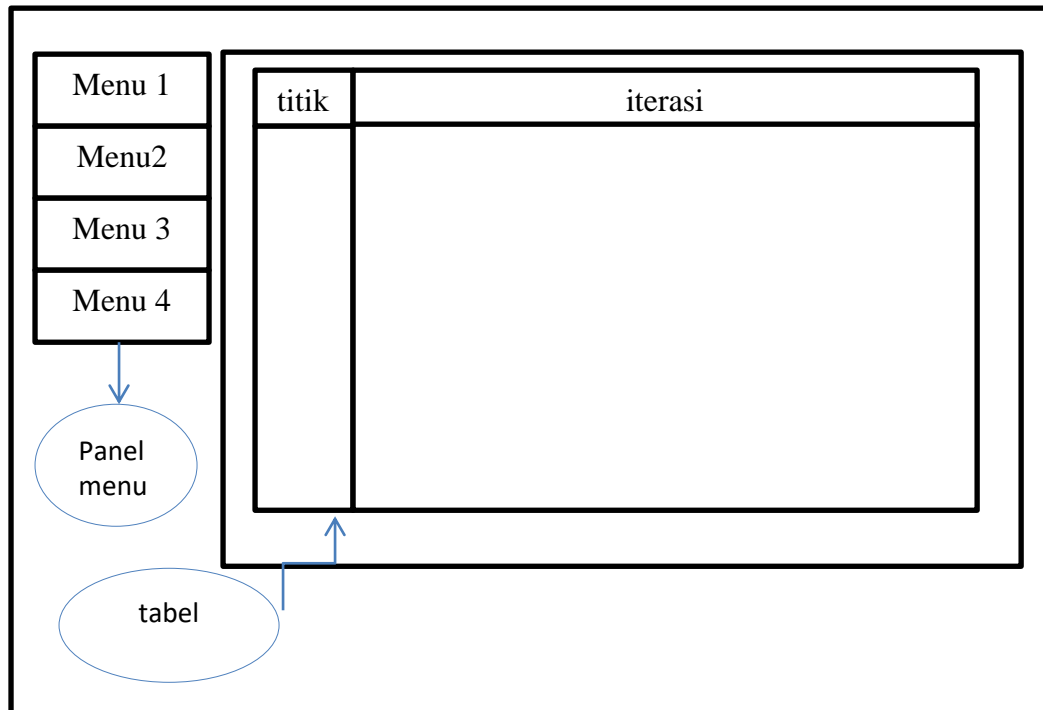
dias ditunjukkan pada Tabel 4.16.

**Tabel 4. 16 penjelasan Antarmuka Sum Responsibility dan Availability**

nama	penjelasan
Panel menu	Terdapat berbagai pilihan menu pada aplikasi dan saling terhubung untuk melihat proses aplikasi.
Tabel	Digunakan untuk menampilkan semua hasil <i>Sum Responsibility</i> dan <i>Availability</i> tiap titik data yang akan diolah pada aplikasi.

#### 4.4.6 Perancangan Antarmuka Iterasi

Perancangan antarmuka pada Gambar 4.10 ditujukan untuk menampilkan hasil dari proses iterasi pada metode *Affinity Propagation*.



**Gambar 4. 10 Rancang Antarmuka Iterasi**

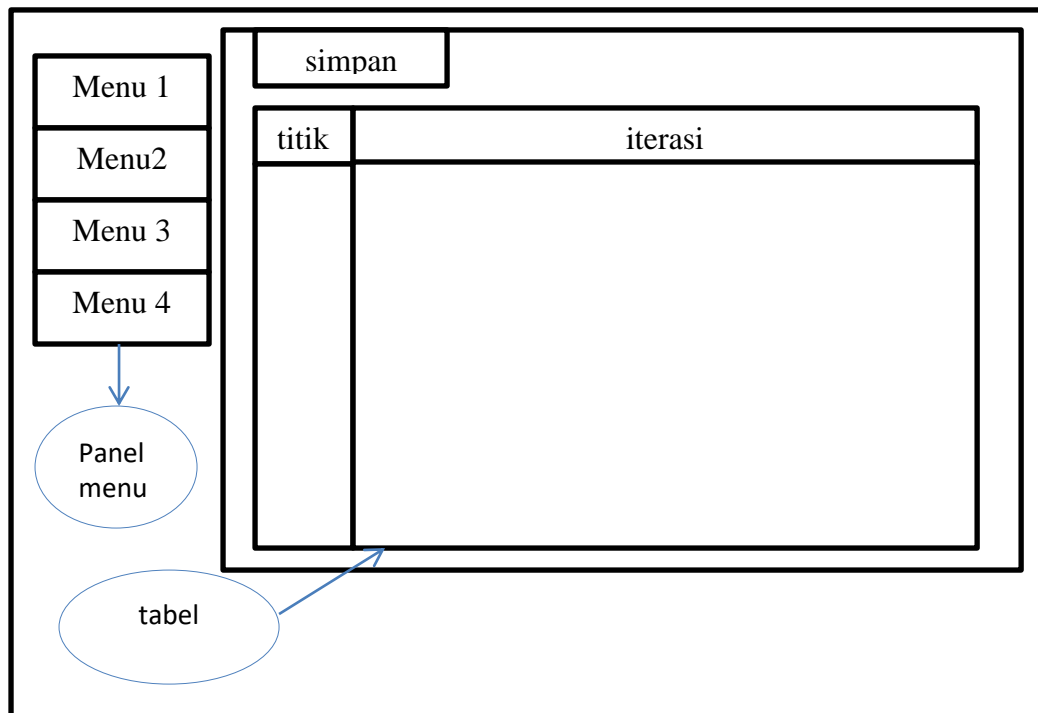
Penjelasan rancangan antarmuka iterasi diatas ditunjukkan pada Tabel 4.17.

**Tabel 4. 17 Penjelasan Antarmuka Iterasi**

nama	penjelasan
Panel menu	Terdapat berbagai pilihan menu pada aplikasi dan saling terhubung untuk melihat proses aplikasi.
Tabel	Digunakan untuk menampilkan semua hasil iterasi tiap titik data yang akan diolah pada aplikasi.

#### 4.4.7 Perancangan Antarmuka *Cluster*

Perancangan antarmuka pada Gambar 4.11 ditujukan untuk menampilkan hasil dari proses *cluster* pada metode *Affinity Propagation*.



**Gambar 4. 11 Rancang Antarmuka *Cluster***

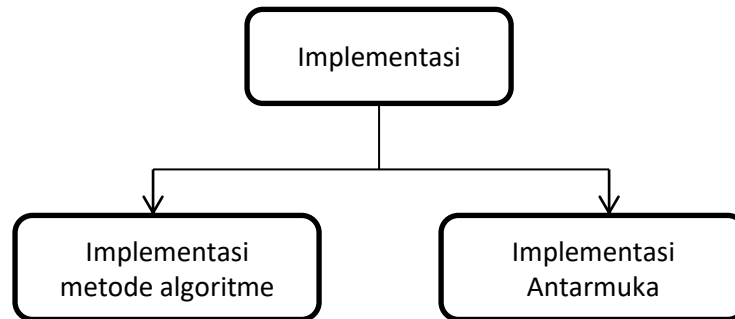
Penjelasan rancangan antarmuka *cluster* diatas ditunjukkan pada Tabel 4.18.

**Tabel 4. 18 Penjelasan Rancang Antarmuka *Cluster***

nama	penjelasan
Panel menu	Terdapat berbagai pilihan menu pada aplikasi dan saling terhubung untuk melihat proses aplikasi.
Menu simpan	Digunakan untuk menyimpan data <i>cluster</i> ke <i>database</i> .
Tabel	Digunakan untuk menampilkan semua hasil <i>Cluster</i> tiap titik data yang akan diolah pada aplikasi.

## BAB 5 IMPLEMENTASI

Pada tahapan implementasi terdiri dari implementasi metode algoritme dan implmentasi antarmuka. Pada Gambar 5.1 di bawah ini merupakan model implmentasi sistem yang digunakan pada penelitian ini:



Gambar 5. 1 Diagram Implmentasi

### 5.1 Implementasi Algoritme

Pada implentasi algoritme akan dibahas metode *Affinity Propagation* yang memiliki proses sebagai berikut.

#### 5.1.1 Proses Algoritme Similarity

Pada proses ini menjelaskan algoritme *Similarity* yang digunakan untuk menghitung kemiripan antar data menggunakan *Euclidean Distance*. Data yang digunakan adalah data titik panas berupa *brightness* dan FRP . *Source code similarity* ditunjukkan pada kode program 5.1.

```
1      for($j = 0; $j<count($list_data); $j++) {  
2          for($i = 0; $i<count($list_data); $i++){  
3              if($i==$j) continue;  
4  
5              $similarity = -sqrt(pow(($list_data[$i]['brightness'] -  
6                  $list_data[$j]['brightness']), 2) +  
7                  pow(($list_data[$i]['frp'] - $list_data[$j]['frp']), 2));  
8  
9              $n[$i][] = $similarity;  
10             $t[$i][] = $list_data[$i]['titik'];  
11             ".$list_data[$j]['titik'];  
12             /* titik self respon */  
13             $p = $list_data[$i]['titik'];  
14             ".$list_data[$j]['titik'];  
15  
16             $maxed[$i][$p] = $similarity;  
17         } }
```

Kode Program 5. 1 kode program similarity

### 5.1.2 Proses Algoritme Responsibility

Pada proses ini menjelaskan algoritme *responsibility* yang digunakan untuk menghitung seberapa cocok suatu data bisa menjadi eksemplar atau pusat data. Data yang digunakan adalah data hasil proses *similarity*. *Source code responsibility* ditunjukkan pada Kode Program 5.2.

```

1      $t_similarity = similarity($list_data, "titik");
2      $similarity = similarity($list_data, "similarity");
3
4      /* for iteration */
5      if($newest_availability != NULL) {
6          $max_av = max(array_keys($newest_availability));
7          $availability = $newest_availability;
8      } else {
9          $availability = availability($list_data, "availability");
10     }
11
12
13     $self_availab = availability($list_data, "selfAvailability");
14     $t = responsibility($list_data, "titik");
15
16     /* sort self availability */
17     ksort($self_availab);
18     foreach($self_availab as $row){
19         $col_selffav[] = $row["values"];
20     }
21
22     /* gabung array similarty n titik similarity */
23     for($a = 0; $a < count($similarity); $a++) {
24         for($b = 0; $b < count($similarity[$a]); $b++) {
25             $temp_similarity[$a][$b] = array("titik" => $t_similarity[$a][$b],
26 "values" => $similarity[$a][$b]);
27         }
28     }
29
30     for($a = 0; $a < count($temp_similarity); $a++) {
31         for($b = 0; $b < count($temp_similarity[$a]); $b++) {
32
33             /* set alias */
34             $r = $temp_similarity[$a][$b];
35             $s = $temp_similarity[$a];
36             $q = $availability[$a][$b];
37             $k = $availability[$a];
38
39             /* search $r in array $s if true then unset array */
40             if(($key = array_search($r, $s)) !== FALSE ) {
41                 unset($s[$key]);
42             }
43             if(($key1 = array_search($q, $k)) !== FALSE ) {
44                 unset($k[$key1]);
45             }
46

```



47	<code>\$col_sim[] = array_column(\$s, "values");</code>
48	<code>\$col_ava[] = array_column(\$k, "values");</code>
49	<code>}</code>
50	<code>}</code>
51	<code>}</code>
52	<code>}</code>
53	<code>/* max column */</code>
54	<code>/* max column */</code>
55	<code>/* max column */</code>
56	<code>/* max column */</code>
57	<code>for(\$a = 0; \$a &lt; count(\$col_ava); \$a++) {</code>
58	<code>for(\$b = 0; \$b &lt; count(\$col_ava[\$a]); \$b++) {</code>
59	<code>\$max[\$a][\$b] = \$col_sim[\$a][\$b] + \$col_ava[\$a][\$b];</code>
60	<code>\$max[\$a][\$b] = \$col_sim[\$a][\$b] + \$col_ava[\$a][\$b];</code>
61	<code>}</code>
62	<code>/* get max column between sim and av */</code>
63	<code>\$sim_av[\$a] = max(\$max[\$a]);</code>
64	<code>\$sim_av[\$a] = max(\$max[\$a]);</code>
65	<code>/* convert array to multidimensional based on temp_similarity data</code>
66	<code>-1 */</code>
67	<code>\$maxed = array_chunk(\$sim_av, count(\$temp_similarity)-1);</code>
68	<code>\$maxed = array_chunk(\$sim_av, count(\$temp_similarity)-1);</code>
69	<code>\$maxed = array_chunk(\$sim_av, count(\$temp_similarity)-1);</code>
70	<code>for(\$i = 0; \$i &lt; count(\$maxed); \$i++) {</code>
71	<code>for(\$j = 0; \$j &lt; count(\$maxed[\$i]); \$j++) {</code>
72	<code>\$maximum[\$i][\$j] = max(\$maxed[\$i][\$j], \$col_selfav[\$i]);</code>
73	<code>\$maximum[\$i][\$j] = max(\$maxed[\$i][\$j], \$col_selfav[\$i]);</code>
74	<code>\$maximum[\$i][\$j] = max(\$maxed[\$i][\$j], \$col_selfav[\$i]);</code>
75	<code>for(\$i = 0; \$i &lt; count(\$temp_similarity); \$i++) {</code>
76	<code>for(\$j = 0; \$j &lt; count(\$temp_similarity[\$i]); \$j++) {</code>
77	<code>\$sum1[\$i][\$j] = \$temp_similarity[\$i][\$j]["values"] -</code>
78	<code>\$maximum[\$i][\$j];</code>
79	<code>\$sum[\$i][\$j] = \$sum1[\$i][\$j];</code>
80	<code>\$titik = \$temp_similarity[\$i][\$j];</code>
81	<code>\$temp[\$i][\$j] = array("titik" =&gt; \$titik, "values" =&gt; \$sum[\$i][\$j]);</code>
82	<code>\$temp[\$i][\$j] = array("titik" =&gt; \$titik, "values" =&gt; \$sum[\$i][\$j]);</code>
83	<code>\$temp[\$i][\$j] = array("titik" =&gt; \$titik, "values" =&gt; \$sum[\$i][\$j]);</code>

**Kode Program 5. 2 kode program responsibility**

### 5.1.3 Proses Algoritme Availability

Pada proses ini menjelaskan algoritme *availability* yang digunakan untuk menghitung seberapa cocok suatu data bisa menjadi anggota dari eksemplar. Data yang digunakan adalah data hasil proses *similarity* dan *responsibility*. *Source code availability* ditunjukkan pada Kode Program 5.3.

1	<code>\$t_similarity = similarity(\$list_data, "titik");</code>
2	<code>\$self_responsi = similarity(\$list_data, "selfResponsibility");</code>
3	<code>\$max_responsi = formattedNewRes(\$list_data, "all");</code>
4	<code>\$max_responsi = formattedNewRes(\$list_data, "all");</code>
5	<code>if(\$newest_responsibility != NULL) {</code>
6	<code>\$responsibility = \$newest_responsibility;</code>
7	<code>} else {</code>

8	\$responsibility = new_responsibility(\$list_data, "newres");
9	}
10	
11	/* separate titik, take only last part */
12	\$titik = "A";
13	for(\$i = 0; \$i < count(\$t_similarity); \$i++) {
14	for(\$j = 0; \$j < count(\$t_similarity[\$i]); \$j++) {
15	\$availability[\$i] = str_replace(\$titik." <code>-></code> ", "",
16	\$t_similarity[\$i]);
17	}
18	\$titik++;
19	}
20	
21	/*merubah bentuk array flatten self responsibility array to match
22	format for easy calculation */
23	for(\$k = 0; \$k < count(\$self_responsi); \$k++) {
24	for(\$l = 0; \$l < count(\$self_responsi[\$k]); \$l++) {
25	\$t = preg_replace("/[\s]<code>(.*?)</code>[\s][A-Z]/", "",
26	\$self_responsi[\$k]["titik"]);
27	\$new_selfrespon[\$t] = 0 - \$self_responsi[\$k]["values"];
28	}
29	}
30	
31	/* count availability */
32	
33	for(\$a = 0; \$a < count(\$availability); \$a++) {
34	for(\$b = 0; \$b < count(\$availability[\$a]); \$b++) {
35	\$values = \$new_selfrespon[\$availability[\$a][\$b]] +
36	max(array(\$new_selfrespon[\$availability[\$a][\$b]],
37	\$max_responsi[\$availability[\$a][\$b]]);
38	\$titik = \$t_similarity[\$a][\$b];
39	\$temp[\$a][\$b] = array("titik" => \$titik, "values" => \$values);
40	}
41	}
42	/*merubah bentuk array*/
43	for(\$a = 0; \$a < count(\$availability); \$a++) {
44	for(\$b = 0; \$b < count(\$availability[\$a]); \$b++) {
45	\$g[\$availability[\$a][\$b]] = array(
46	"titik" => \$availability[\$a][\$b],
47	"values" =>
48	max(array(\$new_selfrespon[\$availability[\$a][\$b]],
49	\$max_responsi[\$availability[\$a][\$b]]));
50	}
51	}

Kode Program 5. 3 kode program availability

#### 5.1.4 Proses Algoritme Penjumlahan Responsibility dan Availability

Pada proses ini menjelaskan algoritme *Sum Responsibility* dan *Availability* yang digunakan untuk menghitung kombinasi antara *Responsibility* dan *Availability*. Data yang digunakan adalah data proses hasil *Responsibility* dan *Availability*. *Source code Sum Responsibility* dan *Availability* ditunjukan pada kode program 5.4.

1	\$t_similarity = similarity(\$list_data, "titik");
2	
3	\$selfavaila = availability(\$list_data, "selfAvailability");
4	\$selfresponsi = similarity(\$list_data, "selfResponsibility");
5	
6	if(\$newest_availability != NULL && \$newest_responsibility != NULL ) {
7	/* get last values of data */
8	\$new_availability = \$newest_availability;
9	\$new_responsibility = \$newest_responsibility;
10	
11	} else {
12	\$new_availability = new_availability(\$list_data, "new_availability");
13	\$new_responsibility = new_responsibility(\$list_data, "newres");
14	}
15	
16	
17	for(\$i = 0; \$i<count(\$new_availability); \$i++) {
18	for(\$j = 0; \$j<count(\$new_availability[\$i]); \$j++) {
19	\$new_resav[\$i][\$j] = array(
20	"titik" => \$new_availability[\$i][\$j]["titik"],
21	"values" => \$new_availability[\$i][\$j]["values"] +
22	\$new_responsibility[\$i][\$j]
23	);
24	}
25	}

**Kode Program 5. 4 Source Code Sum Responsibility Availability**

### 5.1.5 Proses Algoritme Pengelompokan *Cluster*

Pada proses ini menjelaskan algoritme *cluster* dimana semua data hasil iterasi akan dimasukan kedalam *cluster* sesuai dengan proses iterasi. Data yang digunakan adalah data proses hasil iterasi. *Source code cluster* ditunjukkan pada Kode Program 5. 5.

1	function <i>cluster</i> (\$list_data) {
2	\$hasiliter = Iterasi(\$list_data, "data");
3	\$missing_point = Iterasi(\$list_data, "mpoint");
4	
5	for(\$i = 0; \$i<count(\$hasiliter); \$i++) {
6	for(\$j = 0; \$j<count(\$hasiliter[\$i]); \$j++) {
7	\$new_titik[\$i][\$hasiliter[\$i][\$j]["titik']] =
8	\$hasiliter[\$i][\$j]["values"];
9	}
10	
11	/* find array maxed values */
12	\$m[\$i] = max(\$new_titik[\$i]);
13	/* find array maxed keys */
14	\$k[\$i] = array_keys(\$new_titik[\$i], \$m[\$i]);
15	
16	\$data[] = array(
17	"t_a" => preg_replace("/(.*) <code>(.*?)</code>[\s]/", "",

18	<code>\$k[\$i][0]) /* hapus titik depan */,</code>
19	<code>"t_b" =&gt; preg_replace("/[\s]&lt;code&gt;(.*?)&lt;/code&gt; (.*?)", "",</code>
20	<code>\$k[\$i][0]) /* hapus titik belakang */,</code>
21	<code>"values" =&gt; \$m[\$i],</code>
22	<code>);</code>
23	
24	<code>\$titik = preg_replace("/(.*?) &lt;code&gt;(.*?)&lt;/code&gt;[\s]/", "", \$k[\$i][0]);</code>
25	
26	<code>\$data_cluster[ \$titik ][ \$k[\$i][0] ] = \$m[\$i];</code>
27	
28	<code>\$data_t[\$k[\$i][0]] = \$m[\$i];</code>
29	<code>}</code>
30	
31	
32	<code>for(\$i = 0; \$i&lt;count(\$data); \$i++) {</code>
33	<code>\$rotated = \$data[\$i]['t_a']." &lt;code&gt;-&gt;&lt;/code&gt; ".\$data[\$i]['t_b'];</code>
34	<code>\$normal = \$data[\$i]['t_b']." &lt;code&gt;-&gt;&lt;/code&gt; ".\$data[\$i]['t_a'];</code>
35	<code>if(isset(\$data_t[\$rotated]))</code>
36	<code>{</code>
37	<code>if(\$data_t[\$rotated] == \$data_t[\$normal])</code>
38	<code>{</code>
39	<code>\$count_cluster_a = count(\$data_cluster[ \$data[\$i]['t_a'] ]);</code>
40	<code>\$count_cluster_b = count(\$data_cluster[ \$data[\$i]['t_b'] ]);</code>
41	
42	<code>if(\$count_cluster_a &gt; \$count_cluster_b)</code>
43	<code>{</code>
44	<code>unset(\$data_cluster[ \$data[\$i]['t_b'] ]);</code>
45	<code>}</code>
46	
47	<code>}</code>
48	<code>elseif(\$data_t[\$rotated] &lt; \$data_t[\$normal])</code>
49	<code>{</code>
50	<code>unset(\$data_cluster[ \$data[\$i]['t_b'] ]);</code>
51	<code>}</code>
52	<code>}</code>
53	
54	<code>}</code>
55	
56	
57	<code>\$availability = availability(\$list_data, "selfAvailability");</code>
58	
59	<code>\$mp = array_values(\$availability);</code>
60	
61	<code>foreach(array_values(\$data_cluster) as \$key =&gt; \$value) {</code>
62	<code>foreach(\$value as \$k =&gt; \$v) {</code>
63	<code>\$end_point = preg_replace("/(.*?) &lt;code&gt;(.*?)&lt;/code&gt;[\s]/", "",</code>
64	<code>\$k);</code>
65	<code>\$ep[] = \$end_point;</code>
66	<code>}</code>
67	<code>}</code>
68	
69	<code>\$ep = array_unique(\$ep);</code>
70	
71	<code>foreach(\$mp as \$mkey =&gt; \$mvalue) {</code>
72	<code>\$counter = 0;</code>

73	foreach(\$ep as \$k => \$v) {
74	if(\$v == \$mvalue["titik"]) {
75	\$data_cluster[\$mvalue["titik"]][\$mvalue["titik"]."
76	></code> ".\$mvalue["titik"]] = \$mvalue['values'];
77	}
78	\$counter++;
79	}
80	}
81	return \$data_cluster ;
82	}
83	function completeCluster(\$point, \$returned){
84	/* get from db */
85	\$ci =& get_instance();
86	\$list_data = \$ci->db->get("sample125")->result_array();
87	/* extract real point */
88	\$match_point = preg_replace("/[\s]<code>(.*?)</code>(.*?)"/, "",
89	\$point);
90	
91	for(\$i = 0; \$i<count(\$list_data); \$i++) {
92	if(\$match_point == \$list_data[\$i]['titik']) {
93	switch(\$returned) {
94	case "brightness":
95	return \$list_data[\$i]['brightness'];
96	break;
97	case "frp":
98	return \$list_data[\$i]['frp'];
99	break;
100	case "confidence":
101	return \$list_data[\$i]['confidence'];
102	break;
103	}}}

Kode Program 5. 5 kode program pembentukan *cluster*

### 5.1.6 Proses Algoritme Silhouette Coefficient

Proses ini membahas kualitas *cluster* dengan menggunakan metode *Silhouette Coefficient*. Data yang digunakan adalah data proses hasil *cluster*. *Source code Silhouette Coefficient* ditunjukan pada Kode Program 5. 6.

1	for(\$i=0;\$i<count(\$nilai);\$i++){
2	for(\$j=0;\$j<count(\$nilai);\$j++){
3	
4	\$norB = pow((\$nilai[\$i]['br']-\$nilai[\$j]['br']),2);
5	\$norF = pow((\$nilai[\$i]['fr']-\$nilai[\$j]['fr']),2);
6	\$hasil[\$i][\$j] = sqrt(\$norB+\$norF);
7	}
8	}
9	for(\$i=0;\$i<\$jumcluster;\$i++){
10	\$jum[\$i]=0;
11	}
12	
13	for(\$i=0;\$i<\$jumcluster;\$i++){

14	for(\$j=0;\$j< count(\$nilai) ;\$j++){
15	if(\$cluster[\$j]== (\$i+1)){
16	\$jum[\$i]++;
17	}
18	}
19	}
20	
21	for(\$i=0;\$i< count(\$nilai) ;\$i++){
22	\$aai[\$i]=0;
23	\$cluster_i=\$cluster[\$i];
24	\$index_b=0;
25	for(\$j=1;\$j<=count(\$jum);\$j++){
26	\$bi[\$i][\$index_b]=0;
27	for(\$k=0;\$k< count(\$nilai) ;\$k++){
28	if(\$j == \$cluster[\$k] && \$cluster_i==\$cluster[\$k]){
29	\$aai[\$i]+=\$hasil[\$i][\$k];
30	}else if(\$j == \$cluster[\$k] && \$cluster_i != \$cluster[\$k]){
31	\$bi[\$i][\$index_b]+=\$hasil[\$i][\$k];
32	}
33	}
34	
35	
36	if(\$bi[\$i][\$index_b]!= 0 ){
37	\$bi[\$i][\$index_b]/=\$jum[\$j-1];
38	\$index_b++;
39	}else{
40	if(\$jum[\$j-1]>1){
41	\$aai[\$i]/=\$jum[\$j-1];
42	}
43	}
44	}
45	sort(\$bi[\$i], SORT_NUMERIC);
46	\$n = array_shift( \$bi[\$i] );
47	foreach(\$bi as \$k) {
48	if(\$n == 0){
49	\$n = array_shift( \$bi[\$i] );
50	}
51	}
52	
53	\$bix[]=\$n;
54	\$si[\$i]=(\$bix[\$i]-\$aai[\$i])/max(\$bix[\$i],\$aai[\$i]);
55	\$hasilsi = array_sum(\$si)/count(\$si);
56	
57	
58	}
59	return \$hasilsi;
60	}

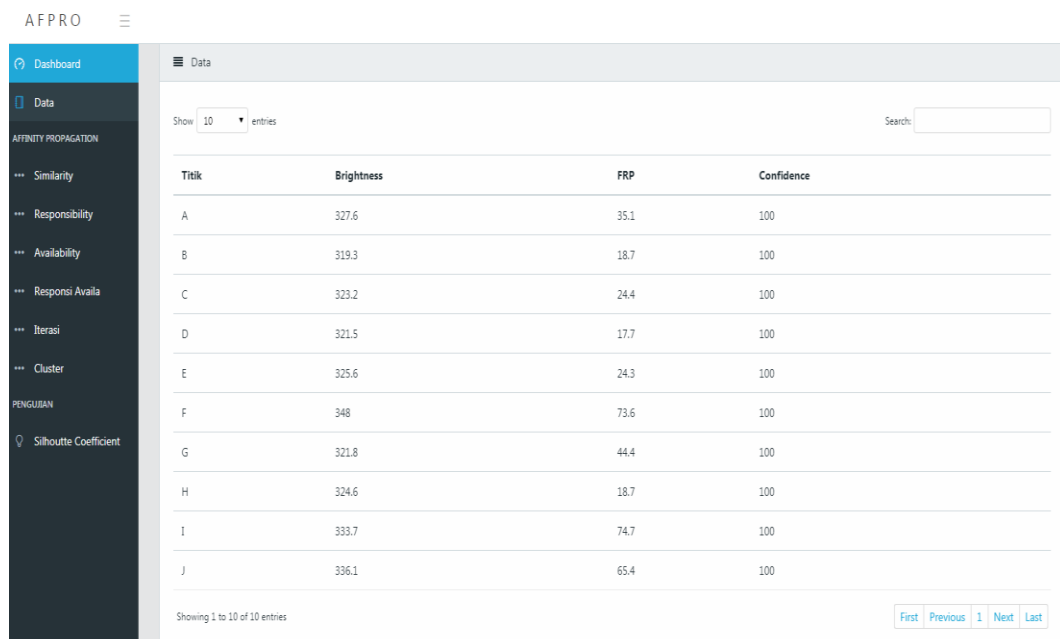
**Kode Program 5. 6 kode program Silhouette Coefficient**

## 5.2 Implementasi Antarmuka

Pada sub bab ini akan dibahas implementasi antarmuka dalam bentuk sebuah aplikasi web yang menggunakan bahasa pemrograman PHP.

### 5.2.1 Implementasi Antarmuka Data

Pada implementasi antarmuka data terdapat halaman yang menunjukkan data titik panas bumi dengan fitur titik, *brightness*, *confidence*, dan FRP. hasil antarmuka data pada bab 4 Gambar 4.4 setelah diimplmentasikan ditunjukan pada Gambar 5.2.



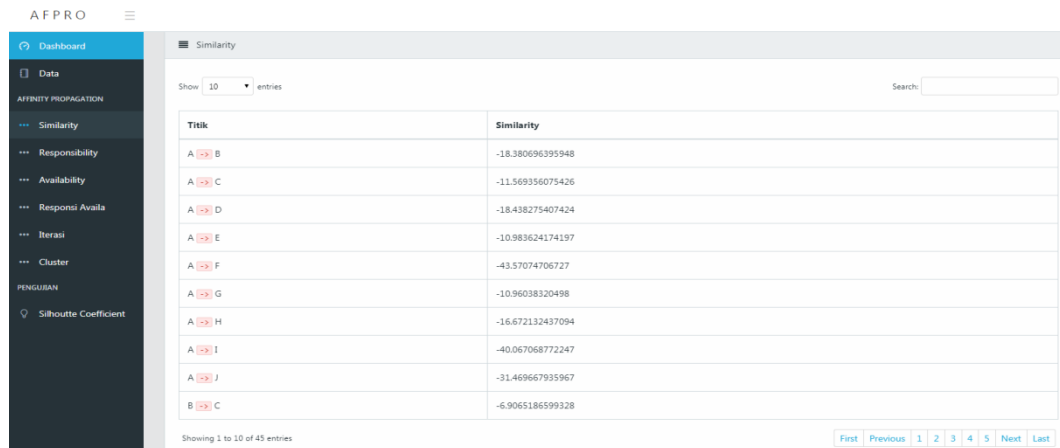
The screenshot displays the AFPRO web application interface. On the left is a dark sidebar with a menu containing 'Dashboard', 'Data', and several sub-items under 'AFFINITY PROPAGATION' and 'PENGUBAN'. The main content area is titled 'Data' and features a table with 10 entries. Above the table, there is a 'Show 10 entries' dropdown and a search bar. The table has four columns: 'Titik', 'Brightness', 'FRP', and 'Confidence'. The data rows are labeled A through J. At the bottom of the table, it says 'Showing 1 to 10 of 10 entries' and there are pagination controls: 'First', 'Previous', '1', 'Next', and 'Last'.

Titik	Brightness	FRP	Confidence
A	327.6	35.1	100
B	319.3	18.7	100
C	323.2	24.4	100
D	321.5	17.7	100
E	325.6	24.3	100
F	348	73.6	100
G	321.8	44.4	100
H	324.6	18.7	100
I	333.7	74.7	100
J	336.1	65.4	100

Gambar 5. 2 Tampilan Halaman Data

## 5.2.2 Implementasi Antarmuka Similarity

Pada implementasi antarmuka *Similarity* terdapat halaman yang menunjukkan data titik panas bumi yang telah diolah menjadi *similarity*. hasil antarmuka *similarity* pada bab 4 Gambar 4.5 setelah diimplmentasikan ditunjukan pada Gambar 5.3



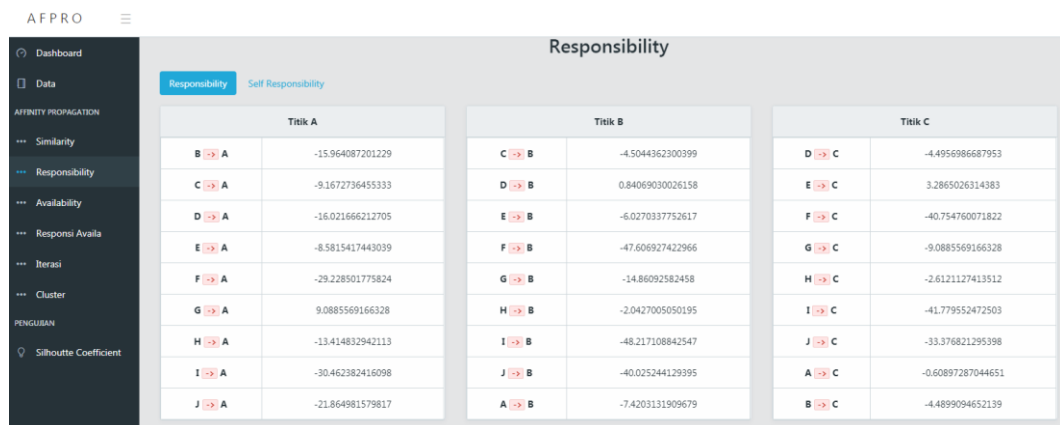
The screenshot shows the AFPRO web application interface. The left sidebar contains navigation options: Dashboard, Data, AFFINITY PROPAGATION (Similarity, Responsibility, Availability, Responsi Availa, Iterasi, Cluster), PENGUBAN, and Silhouette Coefficient. The main content area is titled 'Similarity' and displays a table with two columns: 'Titik' and 'Similarity'. The table lists similarity values for point A compared to points B through J, and for point B compared to point C. The values are negative, indicating dissimilarity. The table is paginated, showing 1 to 10 of 43 entries.

Titik	Similarity
A → B	-18.380696395948
A → C	-11.569356075426
A → D	-18.438275407424
A → E	-10.983624174197
A → F	-43.57074706727
A → G	-10.96038320498
A → H	-16.672132437094
A → I	-40.067068772247
A → J	-31.469667935967
B → C	-6.9065186599328

Gambar 5. 3 Tampilan Halaman Similarity

## 5.2.3 Implementasi Antarmuka Responsibility

Pada implementasi antarmuka *responsibility* terdapat halaman yang menunjukan nilai *responsibility* tiap titik data. hasil antarmuka *responsibility* pada bab 4 Gambar 4.6 setelah diimplmentasikan ditunjukan pada Gambar 5.4.



The screenshot shows the AFPRO web application interface. The left sidebar is the same as in the previous image. The main content area is titled 'Responsibility' and displays three tables side-by-side, each representing the responsibility values for a specific point (A, B, and C). Each table has two columns: 'Titik' and 'Responsibility'. The values are negative, indicating responsibility. The tables are paginated, showing 1 to 10 of 43 entries.

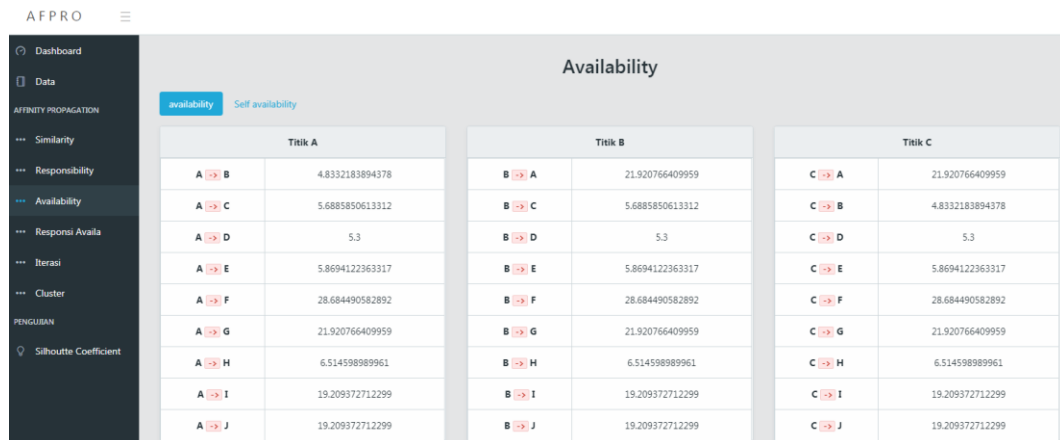
Titik A		Titik B		Titik C	
B → A	-15.964087201229	C → B	-4.5044362300399	D → C	-4.4956986687953
C → A	-9.1672736455333	D → B	0.84069030026158	E → C	3.2865026314383
D → A	-16.021666212705	E → B	-6.0270337752617	F → C	-40.754760071822
E → A	-8.5815417443039	F → B	-47.606927422966	G → C	-9.0885569166328
F → A	-29.228501775824	G → B	-14.86092582458	H → C	-2.6121127413512
G → A	9.0885569166328	H → B	-2.0427005050195	I → C	-41.779552472503
H → A	-13.414832942113	I → B	-48.217108842547	J → C	-33.376821295398
I → A	-30.462382416098	J → B	-40.025244129395	A → C	-0.60897287044651
J → A	-21.864981579817	A → B	-7.4203131909679	B → C	-4.4899094652139

Gambar 5. 4 Tampilan Halaman Responsibility



## 5.2.4 Implementasi Antarmuka Availability

Pada implementasi antarmuka *availability* terdapat halaman yang menunjukkan nilai *availability* tiap titik data. hasil antarmuka *availability* pada bab 4 Gambar 4.7 setelah diimplmentasikan ditunjukan pada Gambar 5.5.



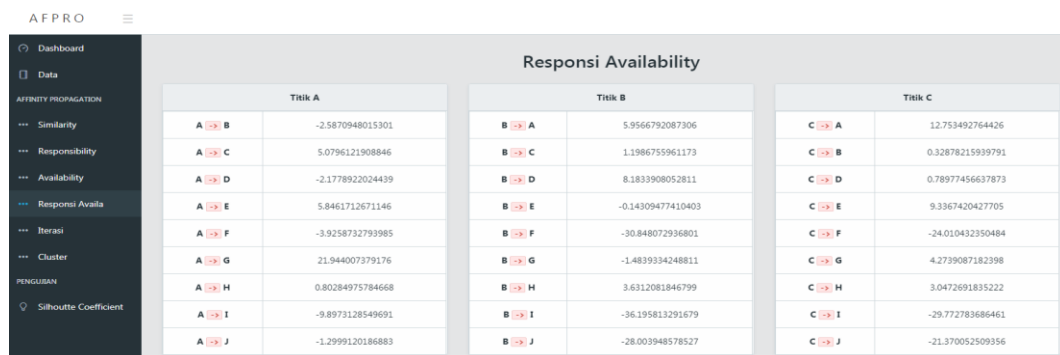
The screenshot shows the AFPRO interface with a sidebar menu on the left. The main content area is titled 'Availability' and contains three tables labeled 'Titik A', 'Titik B', and 'Titik C'. Each table has 10 rows of data, with columns for a point identifier (A-J) and a numerical value. The 'availability' tab is selected in the top navigation bar.

Titik A		Titik B		Titik C	
A → B	4.8332183894378	B → A	21.920766409959	C → A	21.920766409959
A → C	5.6885850613312	B → C	5.6885850613312	C → B	4.8332183894378
A → D	5.3	B → D	5.3	C → D	5.3
A → E	5.8694122363317	B → E	5.8694122363317	C → E	5.8694122363317
A → F	28.684490582892	B → F	28.684490582892	C → F	28.684490582892
A → G	21.920766409959	B → G	21.920766409959	C → G	21.920766409959
A → H	6.514598989961	B → H	6.514598989961	C → H	6.514598989961
A → I	19.209372712299	B → I	19.209372712299	C → I	19.209372712299
A → J	19.209372712299	B → J	19.209372712299	C → J	19.209372712299

Gambar 5. 5 Tampilan Halaman Availability

## 5.2.5 Implementasi Antarmuka Sum Responsibility Availability

Pada implementasi antarmuka *sum responsibility* dan *availability* terdapat halaman yang menunjukkan nilai *sum responsibility* dan *availability* tiap titik data pada iterasi pertama. hasil antarmuka *sum responsibility* dan *availability* pada bab 4 Gambar 4.8 setelah diimplmentasikan ditunjukan pada Gambar 5.6.



The screenshot shows the AFPRO interface with a sidebar menu on the left. The main content area is titled 'Responsi Availability' and contains three tables labeled 'Titik A', 'Titik B', and 'Titik C'. Each table has 10 rows of data, with columns for a point identifier (A-J) and a numerical value. The 'Responsi Availa' tab is selected in the top navigation bar.

Titik A		Titik B		Titik C	
A → B	-2.5870948015301	B → A	5.9566792087306	C → A	12.753492764426
A → C	5.0796121908846	B → C	1.1986755961173	C → B	0.32878215939791
A → D	-2.1778922024439	B → D	8.1833908052811	C → D	0.78977456637873
A → E	5.8461712671146	B → E	-0.14309477410403	C → E	9.3367420427705
A → F	-3.9258732793985	B → F	-30.848072936801	C → F	-24.010432350484
A → G	21.944007379176	B → G	-1.4839334248811	C → G	4.2739087182398
A → H	0.80284975784668	B → H	3.6312081846799	C → H	3.0472691835222
A → I	-9.8973128549691	B → I	-36.195813291679	C → I	-29.772783686461
A → J	-1.2999120186883	B → J	-28.003948578527	C → J	-21.370052509356

Gambar 5. 6 Tampilan Halaman Responsi Availability

## 5.2.6 Implementasi Antarmuka Iterasi

Pada implementasi antarmuka Iterasi terdapat halaman yang menunjukkan nilai *sum responsibility* dan *availability* dari iterasi terakhir. Hasil implementasi antarmuka halaman iterasi ditunjukkan pada Gambar 5.7.

AFPRO

Dashboard	Iterasi (3)					
Data						
AFFINITY PROPAGATION						
Similarity						
Responsibility						
Availability						
Responsi Availa						
Iterasi						
Cluster						
PENGUBAN						
Silhouette Coefficient						

Titik A			Titik B			Titik C		
A → B		-24.507861211489	B → A		1.1234608192928	C → A		7.0649077030949
A → C		-17.72557442062	B → C		-5.6424238141587	C → B		-12.424710605028
A → D		-24.565440222965	B → D		-1.1234608192928	C → D		-12.43049980861
A → E		-17.139842519391	B → E		-7.1650213593806	C → E		-7.9493279046402
A → F		-25.846639689358	B → F		-36.804752145532	C → F		-36.76392511491
A → G		0	B → G		-7.4406126336117	C → G		-8.4795840461862
A → H		-21.117916652113	B → H		-2.3254710240508	C → H		-9.7062235809038
A → I		-31.818079264928	B → I		-42.15249250041	C → I		-42.526276450887
A → J		-23.220678428648	B → J		-33.960627787258	C → J		-34.123545273782

Gambar 5. 7 Tampilan Halaman Iterasi

## 5.2.7 Implementasi Antarmuka Cluster

Pada implemantasi antarmuka *cluster* terdapat halaman yang menunjukkan jumlah dan anggota *cluster* tiap titik data. hasil antarmuka *cluster* pada bab 4 Gambar 4.9 setelah diimplmentasikan ditunjukkan pada Gambar 5.8.

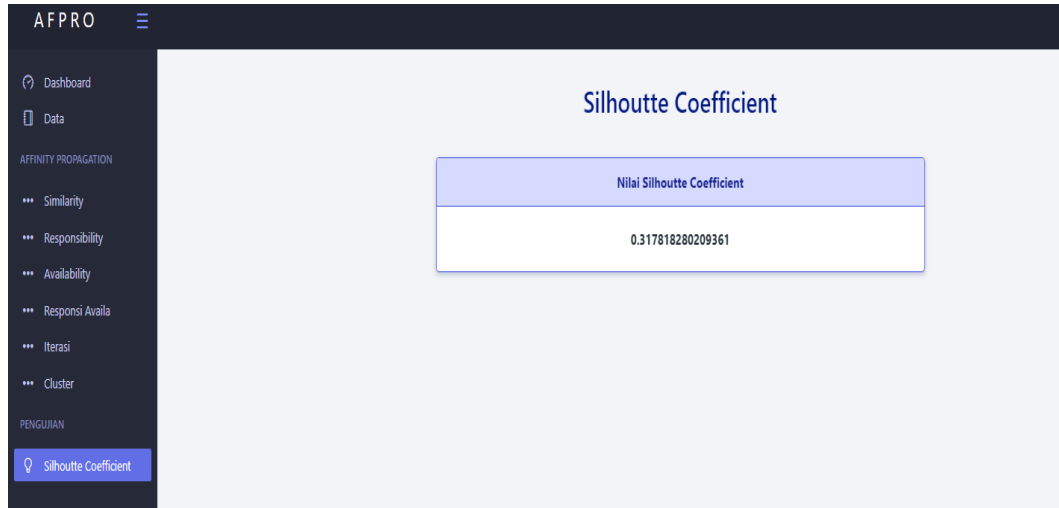
AFPRO

</

Gambar 5. 8 Tampilan Halaman Cluster

## 5.2.8 Implementasi Antarmuka Silhouette Coefficient

Pada implementasi antarmuka *silhouette coefficient* terdapat halaman yang menunjukkan nilai *silhouette coefficient* dari *cluster* yang terbentuk oleh metode *affinity propagation*. Hasil implementasi antarmuka halaman *silhouette coefficient* ditunjukkan pada Gambar 5.9.



Gambar 5. 9 Tampilan Halaman Silhouette Coefficient

## BAB 6 PENGUJIAN

Pengujian yang akan dilakukan pada bab ini adalah pengujian kualitas *cluster* hasil metode *Affintiy Propagation* dengan menggunakan data titik panas bumi dengan jumlah yang berbeda.

### 6.1 Pengujian Kualitas *Cluster* Berdasarkan Jumlah Data

Pengujian kualitas *Cluster* digunakan untuk mengetahui seberapa bagus kualitas sebuah *Cluster* yang terbentuk dari metode *Affinity Propagation* dengan menggunakan metode *Silhouette Coefficient*. Hasil *Clustering* akan dinilai dengan menggunakan *Silhoutte Coefficient* dengan harapan mendapat hasil yang tinggi yaitu nilai yang mendekati angka 1. Pengujian dilakukan dengan cara mengganti jumlah data titik panas yaitu *Brightness* dan *FRP*.

#### 6.1.1 Pengujian *Cluster* Berdasarkan Jumlah Data

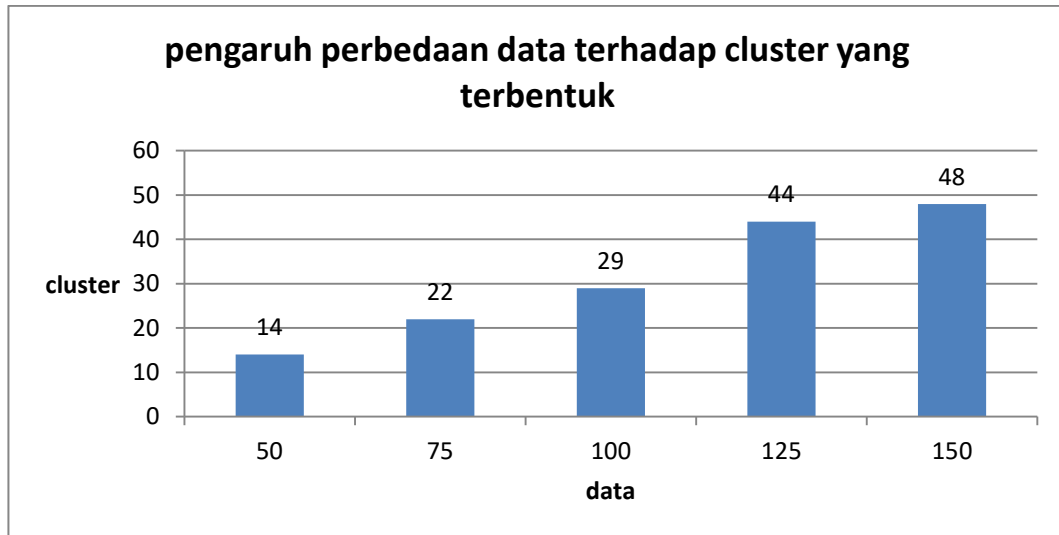
Pengujian nilai *Cluster* berdasarkan jumlah data dilakukan dengan cara memasukan data titik panas dengan jumlah yang berbeda untuk mengetahui apakah jumlah data berpengaruh kepada kualitas *Cluster* dan jumlah *cluster* yang terbentuk. Untuk jumlah data yang digunakan untuk pengujian adalah 50 data, 75 data, 100 data, 125 data, 150 data.

Pada pengujian ini dilakukan dengan memasukan jumlah data 50, 75, 100, 125, 150 untuk mengetahui berapa *cluster* yang terbentuk dan kualitas *cluster* tersebut. Hasil pengujian dapat dilihat pada Tabel 6.1

Tabel 6. 1 pengujian

no	data	Cluster	Silhouette coefficient
1	50	14	0.232208
2	75	22	0.235706
3	100	29	0.214988
4	125	44	0.317818
5	150	48	0.289378

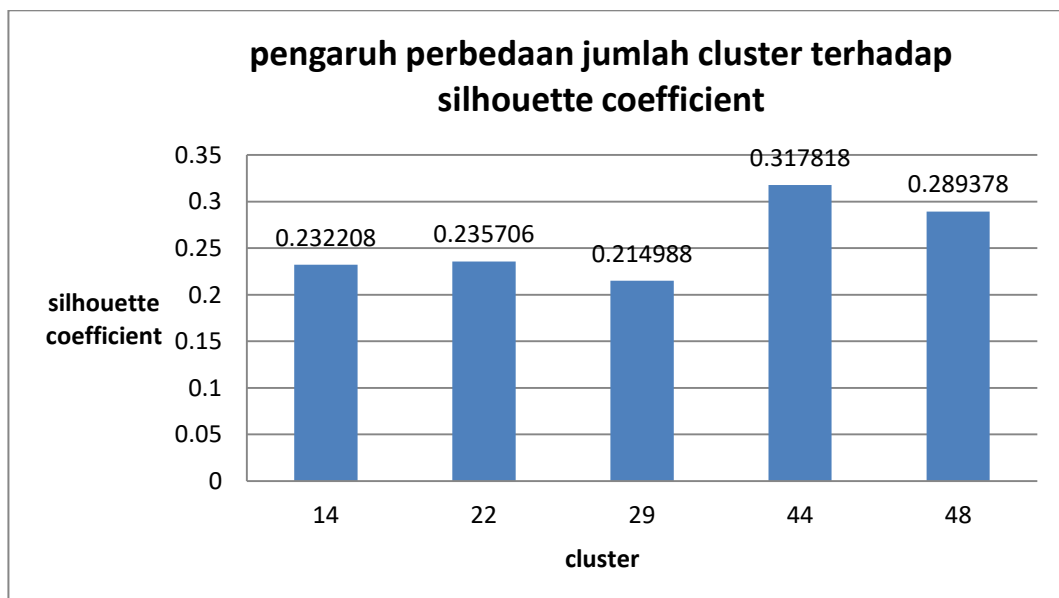
Untuk mengetahui pola dari pengujian antara data dengan *cluster* yang terbentuk dibentuk grafik pengaruh perbedaan data latih terhadap *cluster* yang terbentuk yang dapat dilihat pada Gambar 6.1.



**Gambar 6. 1 Grafik pengaruh perbedaan data latih terhadap *cluster* yang terbentuk.**

Pada Gambar 6.1 menunjukkan perbedaan jumlah data mempengaruhi jumlah *cluster* yang terbentuk. Semakin banyak jumlah data maka semakin banyak pula *cluster* yang terbentuk.

Untuk mengetahui pola dari pengujian antara *cluster* dengan nilai *silhouette coefficient* dibentuk grafik pengaruh perbedaan *cluster* terhadap *silhouette coefficient* yang terbentuk yang dapat dilihat pada Gambar 6.2.



**Gambar 6. 2 grafik pengaruh perbedaan jumlah data terhadap silhouette coefficient**

Pada Gambar 6.2 menunjukkan perbedaan jumlah *cluster* tidak mempengaruhi nilai *silhouette coefficient*. Jumlah *cluster* yang semakin banyak tidak membuat nilai *silhouette coefficient* semakin besar. Pada grafik di atas diketahui hasil tertinggi *silhouette coefficient* didapat pada data 125 dengan *cluster* 44 yaitu dengan nilai *silhouette coefficient* 0.317818.

### 6.1.2 Analisis Pengujian Kualitas *Cluster*

Pada sub bab ini dilakukan proses analisis hasil pengujian kualitas *cluster* yang dilakukan untuk mengetahui pola dari metode *Affinity Propagation*. Perubahan pola yang dihasilkan dari pengaruh data terhadap *silhouette coefficient* adalah jumlah data tidak mempengaruhi besar kecilnya nilai *silhouette coefficient*. tetapi *cluster* yang terbentuk akan lebih banyak jika data yang dimasukkan lebih besar. Dari pengujian yang telah dilakukan diperoleh nilai *silhouette coefficient* tertinggi dari 125 data dengan 44 *cluster* adalah 0.317818 yang terbilang kecil. Hal ini dikarenakan banyaknya jumlah *cluster* yang besar akan membuat jarak data antar *cluster* semakin kecil dan mengakibatkan banyak sekali kemiripan antar data, jadi banyaknya data yang mirip akan membuat nilai *silhouette coefficient* menjadi kecil. Jumlah *cluster* yang besar dikarenakan pada *affinity propagation* tidak ada input untuk menentukan *cluster* sehingga program dengan sendirinya menentukan jumlah *cluster*.

Pengujian ini dilakukan tanpa menggunakan filter fitur *confidence*. *Confidence* adalah tingkat kepercayaan bahwa titik panas yang dipantau oleh satelit berpotensi kebakaran dengan nilai persentase 0% - 100%. Meskipun demikian hasil *cluster* menunjukkan kemiripan dalam fitur *confidence* tiap titik anggota *cluster*. Jadi bisa diasumsikan bahwa *cluster* tertentu yang memiliki rata-rata *confidence* >80% sangat berpotensi kebakaran. Tetapi ada beberapa *cluster* yang memiliki nilai *confidence* antar data sangat jauh atau tidak mirip. Maka dari itu dilakukan penilaian *correlation coefficient* untuk mengetahui hubungan antar fitur.

Untuk mengetahui bagaimana hubungan antar fitur data yaitu *brightness*, FRP, *confidence* apakah saling berpengaruh satu sama lain telah dilakukan perhitungan *correlation coefficient* pada kelompok data dengan nilai *silhouette coefficient* terbaik yaitu 125 data. Hasil perhitungan ditunjukkan pada Tabel 6.2.

**Tabel 6. 2 nilai correlation coefficient**

<i>Brightness</i> terhadap FRP	<i>Brightness</i> terhadap <i>confidence</i>	FRP terhadap <i>confidence</i>
0.829861189	0.288087594	0.184065112

Hasil *correlation coefficient* menunjukkan nilai positif akan tetapi bernilai rendah terhadap *confidence* yang artinya hubungan *brightness* dan FRP terhadap *confidence* tidak cukup kuat. Data 44 *cluster* yang terbentuk dicantumkan pada lampiran.

## BAB 7 PENUTUP

Pada bab penutup akan dibahas kesimpulan dari penelitian yang telah dilakukan dan saran yang diberikan penulis untuk penelitian selanjutnya.

### 7.1 Kesimpulan

Berdasarkan hasil dari pengujian serta analisis dari *clustering* titik panas bumi menggunakan algoritme *affinity propagation* dengan metode pengujian *silhouette coefficient*, sehingga diperoleh kesimpulan sebagai berikut:

1. Metode *Affinity Propagation* dapat diimplementasikan untuk melakukan *clustering* data titik panas bumi dengan hasil *cluster* yang terbentuk dapat dilakukan pemantauan potensi kebakaran dengan cara melihat rata-rata nilai *confidence* tiap *cluster*.
2. Kualitas *cluster* pada metode *Affinity Propagation* didapat nilai pengujian pada 125 data terbentuk 44 *cluster* dengan nilai *silhouette coefficient* 0.317818. nilai tersebut terbilang rendah karena banyaknya jumlah *cluster* yang terbentuk akan membuat jarak antar data semakin kecil dan membuat banyak kemiripan data antar *cluster*.

### 7.2 Saran

Saran yang dapat diberikan untuk pengembang penelitian selanjutnya adalah:

1. Pada penelitian ini *range* data antara fitur *brightness* dengan FRP sangat jauh. Perlu dilakukan normalisasi data sehingga *range* antar data bernilai 0 sampai 1.
2. Memasukan fitur *confidence* untuk dilakukan perhitungan pada algoritme atau melakukan filter *confidence* hanya untuk nilai *confidence* yang besar agar hasil *cluster* lebih mirip pada fitur *confidence*-nya.

## DAFTAR PUSTAKA

- Adzkia, Agmalaro Farisya , Asyhar Muhammad., 2016. *Deteksi Pencilan Kontekstual pada Data Titik Panas di Provinsi Riau menggunakan Algoritme K-Medoids*. Institut Pertanian Bogor.
- Baehaki, Aulia, D., Sitanggang, M., Sukaesih, I., 2014. *Deteksi pencilan data titik api di provinsi riau menggunakan algoritme Clustering K-Means*. Institut Pertanian Bogor.
- Boschetti, L., Justice, C., et al.2008. *global Burned Area Detection : MODIS and Beyond*. [e-junal].
- Cortez, P. 2010. *Data Mining with Neural Networks and Support Vector Machines Using the R/rmine R Tool*. In P. Perner (Ed.), *Advances in Data Mining*, 10th Industrial Conference on Data Mining (ICDM 2010). Berlin.
- Delbert, D., 2009. *Affinity Propagation: Clustering Data by Passing Messages*. Thesis projects. Department of Electrical & Computer Engineering University of Toronto. [e-jurnal]. Tersedia di < <http://www.cs.columbia.edu/~delbert/>> [diakses 5 maret 2016]
- Giglio, L., Descloitres, J., Justice, C.O., & Kaufman, Y. J., 2003. *An enhanced contextual fire detection algorithm for MODIS*. Remote Sensing of Environment.
- Guojun, G., Michael, K.N., 2014. *Subspace Clustering using Affinity Propagation*. [e-jurnal].
- Hassanabadi, B., Shea, C., Zhang, L., Valaee, S., 2013. *Clustering in vehicular Ad Hoc Networks using Affinity Propagation*. [e-jurnal].
- Han, J., Kamber, M., dan Pei, J., 2006. *Data Mining: Concepts andTechniques*. [book].
- Heryalianto, S., C., 2006. *Studi Tentang Sebaran Titik Panas (Hotspot) Sebagai Penduga Kebakaran Hutan dan Lahan di Propinsi Kalimantan Barat Tahun 2003 dan Tahun 2004*. Institut Pertanian Bogor.
- Ke-Lin Du, Swamy, M., N., S., 2013. *Clustering I: Basic Clustering Models and Algorithms*. Neural Networks and Statistical Learning.
- Kruegel, C., Valeur, F., Vigna, G., 2005. *Intrusion Detection and Correlation challenges and Solution*. University of California, Santa Barbara, Springer Science USA.



- M. Bentar, C., 2015. *Deteksi Pencilan pada Titik Panas Menggunakan Clustering berbasis Medoids*.
- Meng, J., Hao, H., Luan, Y., 2015. *Classifier Ensemble Selection based on Affinity Propagation Clustering*. [e-jurnal].
- Peter, J., Rousseeuw, 1987. *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. University of Fribourg, ISES, CH-I 700 Fribourg Switzerland.
- Raharjo, S., Winarko, E., 2014. *KLASTERISASI, KLASIFIKASI DAN PERINGKASAN TEKS BERBAHASA INDONESIA*. Universitas Gunadarma.
- Sigit, P., Hadiwardoyo, 2015. *Implementasi data mining untuk mendukung system manajemen perkerasan jalan di Indonesia*. Jurnal HPJI Vol. 1 No. 2.
- Sitanggang, S., 2013. *Association Rules Mining untuk Data Kebakaran Hutan Menggunakan Algoritme ECLAT dan SPADE*. Departemen Ilmu Komputer, Institut Pertanian Bogor.
- The earth observing system data and information system NASA* <<https://earthdata.nasa.gov/about>> [diakses 30 april 2016].
- Xiangliang, Z., Cyril, F., Michele, S., *Data Streaming with Affinity Propagation*. [e-jurnal]. Tersedia di <<https://mine.kaust.edu.sa/Documents/papers/>> [diakses 5 maret 2016].

## LAMPIRAN 44 *CLUSTER*

Cluster 1			
Point	Br	Fr	Co
B -> BC	316.4	17.8	0
AQ -> BC	311.6	18.1	48
BM -> BC	316.6	17.1	38
CA -> BC	315.8	16.2	33
CB -> BC	315.2	16.8	48
BC -> BC	314.4	19.1	25

Cluster 2			
Point	Br	Fr	Co
C -> Y	313.6	10.7	59
Y -> Y	314.3	10.6	61

Cluster 3			
Point	Br	Fr	Co
E -> M	318.9	13.2	0
M -> M	319.7	12.6	65

Cluster 4			
Point	Br	Fr	Co
F -> AQ	310.6	19.7	23
AQ -> AQ	311.6	18.1	48

Cluster 5			
Point	Br	Fr	Co
I -> CI	315.9	13	63
BY -> CI	315.1	15.1	41
CI -> CI	314.5	13.3	18

Cluster 6			
Point	Br	Fr	Co
J -> A	317	9.4	58
AJ -> A	314.9	8.2	53
A -> A	316.8	7.8	67

Cluster 7			
Point	Br	Fr	Co
K -> DG	327.6	35.1	72
BB -> DG	327.2	35.3	77
DG -> DG	329.4	36.4	81

Cluster 8			
Point	Br	Fr	Co
L -> AF	323.2	25.3	60
R -> AF	323.2	24.4	76
AF -> AF	325.6	24.3	78

Cluster 9			
Point	Br	Fr	Co
M -> CY	319.7	12.6	65
CK -> CY	320.7	10.8	64
DS -> CY	321.6	11	36
CY -> CY	321.5	12.7	74

Cluster 10			
Point	Br	Fr	Co
N -> V	310.9	5.8	32
CX -> V	312.5	6.5	55
V -> V	311.4	6.4	27

Cluster 11			
Point	Br	Fr	Co
O -> D	318.6	17	43
Z -> D	322.5	16.4	0
CY -> D	321.5	12.7	74
D -> D	320.4	15.5	0

Cluster 12			
Point	Br	Fr	Co
Q -> BK	319.3	18.7	71
BW -> BK	322.6	24.2	52
BZ -> BK	319.3	24.1	65
CD -> BK	319.7	19.3	50
DM -> BK	320.7	25.1	0
BK -> BK	321.2	21.8	64

Cluster 13			
Point	Br	Fr	Co
T -> S	315.6	63.2	64
S -> S	316	65.4	65

Cluster 14			
Point	Br	Fr	Co
U -> AJ	314.6	10	37
AI -> AJ	314.6	9.5	57
AJ -> AJ	314.9	8.2	53

Cluster 15			
Point	Br	Fr	Co
W -> Z	321.5	17.7	73
Z -> Z	322.5	16.4	0

Cluster 16			
Point	Br	Fr	Co
AB -> CX	312.6	6.8	55
CX -> CX	312.5	6.5	55

Cluster 17			
Point	Br	Fr	Co
AC -> AG	342	52.1	69
AZ -> AG	336.1	65.4	86
CN -> AG	342.5	51	69
AG -> AG	348	73.6	94

Cluster 18			
Point	Br	Fr	Co
AF -> AD	325.6	24.3	78
CZ -> AD	327.4	20.2	80
AD -> AD	330.1	22.4	59

Cluster 19			
Point	Br	Fr	Co
AG -> BI	348	73.6	94
BL -> BI	340.8	84.4	89
CT -> BI	329.1	92.8	71
BI -> BI	347.1	109.5	93

Cluster 20			
Point	Br	Fr	Co
AK -> CO	318	11.4	0
CG -> CO	317.9	11.4	73
CO -> CO	318	10.5	70

Cluster 21			
Point	Br	Fr	Co
AL -> BY	316.2	15.6	68
BY -> BY	315.1	15.1	41

Cluster 22			
Point	Br	Fr	Co
AM -> F	311.2	21.1	47
F -> F	310.6	19.7	23

Cluster 23			
Point	Br	Fr	Co
AN -> AP	313.1	26.3	57
AX -> AP	314.6	22.1	52
AP -> AP	314.3	24.5	54



Cluster 24			
Point	Br	Fr	Co
AO -> G	313.9	35	59
G -> G	310.5	35.4	0

Cluster 25			
Point	Br	Fr	Co
AT -> BS	315.7	27.5	64
BN -> BS	321.6	30.8	70
BS -> BS	317.6	28.7	58

Cluster 26			
Point	Br	Fr	Co
AU -> AY	330.1	64.7	0
AY -> AY	333.7	74.7	84

Cluster 27			
Point	Br	Fr	Co
AV -> AR	326.6	44.5	54
BG -> AR	323	39.8	73
AR -> AR	321.8	44.4	74

Cluster 28			
Point	Br	Fr	Co
AW -> CZ	324.6	18.7	73
CL -> CZ	324.2	18.3	78
CZ -> CZ	327.4	20.2	80

Cluster 29			
Point	Br	Fr	Co
AY -> BL	333.7	74.7	84
BL -> BL	340.8	84.4	89

Cluster 30			
Point	Br	Fr	Co
BF -> AT	317.5	25.4	41
BR -> AT	316.3	25.4	57
AT -> AT	315.7	27.5	64

Cluster 31			
Point	Br	Fr	Co
BJ -> DL	330.2	53.9	81
DL -> DL	331.4	55.1	0

Cluster 32			
Point	Br	Fr	Co
BO -> BP	328.6	43.7	79
DG -> BP	329.4	36.4	81
BP -> BP	326.8	40.9	68

Cluster 33			
Point	Br	Fr	Co
BU -> BV	317.5	19.9	29
BV -> BV	317.2	21.5	56

Cluster 34			
Point	Br	Fr	Co
BX -> AH	312.7	12.4	28
DH -> AH	312.5	13.3	54
AH -> AH	310.6	13	39

Cluster 35			
Point	Br	Fr	Co
CE -> DN	320	8.4	35
CF -> DN	321.3	8.1	50
DN -> DN	320.1	6.6	36

Cluster 36			
Point	Br	Fr	Co
CJ -> CQ	352.2	192.3	91
CP -> CQ	384.7	234.8	71
CQ -> CQ	379.6	203.3	0

Cluster 37			
Point	Br	Fr	Co
CM -> CR	333.8	33.1	86
DJ -> CR	336.9	31	85
CR -> CR	333.4	28.7	75

Cluster 38			
Point	Br	Fr	Co
CR -> DP	333.4	28.7	75
DD -> DP	335.8	20.1	76
DP -> DP	335.4	25.1	82

Cluster 39			
Point	Br	Fr	Co
CU -> AU	322.4	62.8	68
CV -> AU	321.9	61.1	68
DL -> AU	331.4	55.1	0
AU -> AU	330.1	64.7	0

Cluster 40			
Point	Br	Fr	Co
DA -> CM	337.2	32.8	88
CM -> CM	333.8	33.1	86

Cluster 41			
Point	Br	Fr	Co
DC -> DO	324.7	7.3	60
DO -> DO	326.9	7.9	40

Cluster 42			
Point	Br	Fr	Co
DE -> DQ	328.9	12.5	59
DF -> DQ	327	11.3	59
DU -> DQ	327	12.5	69
DQ -> DQ	329.6	16	65

Cluster 43			
Point	Br	Fr	Co
DK -> CJ	354.4	168.4	73
CJ -> CJ	352.2	192.3	91

Cluster 44			
Point	Br	Fr	Co
DR -> DT	322.8	9.9	40
DT -> DT	324.2	10.2	62